

**A Multi-Agent Crop Production Decision Support System
for Technology Transfer**

**A Thesis Submitted to the College of Graduate Studies and Research
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy
in the College of Graduate Studies and Research
University of Saskatchewan**

**by
Murray J. Bentham
Spring, 2000**

**© Copyright Murray J. Bentham, 2000.
All Rights Reserved**



**National Library
of Canada**

**Acquisitions and
Bibliographic Services**

**395 Wellington Street
Ottawa ON K1A 0N4
Canada**

**Bibliothèque nationale
du Canada**

**Acquisitions et
services bibliographiques**

**395, rue Wellington
Ottawa ON K1A 0N4
Canada**

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-63842-1

Canada

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purpose may be granted by the professor or professors who supervised my thesis work, or in their absence, by the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Request for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Dean of the College of Graduate Studies and Research
University of Saskatchewan
3 Campus Drive
Saskatoon, Saskatchewan
Canada S7N 5A4

Abstract

The purpose of this research was to study agricultural crop production *decision support systems*¹ as a means of transferring agricultural technology from research labs and plots to producers, extension specialists, agriculture service agencies, and scientists on the Western Canadian Prairies. The primary objective was to develop a computer application program that would fulfill the farm manager's decision support needs and be "open" to future enhancements. This interdisciplinary study has a strong agricultural presence in the application context of the resultant computerized agricultural decision support system, with agronomics being the foundation on which the system was built, and computer science being the toolbox used to build it.

Farm Smart 2000 is the resultant decision support system, providing "single-window" access to three different tiers of decision support utilizing the Internet, *expert systems*² and integrated multiple heterogeneous *reusable agents*³ in a cooperative problem-solving environment. Farm Smart 2000 provides support for most management aspects of crop production including variety selection, crop rotations, weed management, disease management, residue management, harvesting, soil conservation, and economics, for the crops of wheat, canola, barley, peas, and flax.

Tier-3, the most sophisticated level of Farm Smart 2000, is the focus of this dissertation and utilizes multiple reusable agents, integrating them such that they cooperate together to solve complex interrelated crop production problems. A Global Control Expert achieves the required communication and coordination among the agents resulting in an "open system", enabling Farm Smart 2000 to extend its problem-solving capabilities by integrating additional agents and knowledge, without system re-engineering, thereby remaining an ongoing technology transfer vehicle.

¹ A *decision support system* is a computer program that analyses problems spanning several knowledge or problem areas producing results that aid the management decision-making process. This is accomplished by combining information, knowledge, and human expertise, through the integration of expert systems (see 2), rule-sets, site-specific data, and any other associated software.

² An *expert system* is a computer program that solves complicated problems, within a specific knowledge or problem area, that would otherwise require human expertise. It simulates or mimics the human expert's reasoning process by applying specific knowledge and thought processes called artificial intelligence.

³ Expert systems integrated with each other within a decision support system are called *agents* (Bond and Gasser, 1988). *Reusable agents* are modular computer programs (e.g. expert systems) which can be used in more than one computer application with little or no modification (Neches et al., 1991).

Acknowledgments

I want to remember the late Dr. Ron Howarth who recognized my desire to further my education and research experience, and who, with the support and approval of Dr. Dorrell, initiated an opportunity for me to do so. I want to especially thank Dr. Wayne Lindwall, my employment supervisor and a member of my Advisory Committee, for his continual encouragement, understanding, and guidance throughout this research project. I wish to extend my thanks to Dr. Bob McKercher for his time and effort in guiding me through the uncharted administrative waters of an interdisciplinary program of studies. I am very grateful to Dr. Ernie Barber, my supervisor, and Dr. Darwin Anderson, the chairman of my Advisory Committee, for accepting the challenge of an interdisciplinary student and “making it all happen” for me. I want to thank Dr. Don Acton and Dr. John Cooke, members of my Advisory Committee, for their support and hard work. I appreciate the effort and contributions of my external examiner, Dr. Marvin Shaffer, USDA-ARS, Colorado State University.

I extend my appreciation to the producers and many people from universities, industry, governments, and Agriculture and Agri-Food Canada research centres for their contributions, without which this research would not have been possible. Henry de Gooijer was always available to give generously of his services, for which I am very thankful. I wish to thank the special faculty, staff, and students in the Colleges of Agriculture and Engineering who graciously provided advice and support.

Most importantly, I wish to thank my family; my wife Lorraine for her never-ending patience, support, and understanding, and my daughters Ricarda and Davida for their sacrifices on my behalf during these years. This work is dedicated to Ricarda and Davida who inspire me with their energy and creativity. You learned at a young age the value of an education, and it is my hope that this wisdom remains with you throughout your lives. Always remember your Grandpa Epp’s tenet: “Your education is yours forever.”.

This research was funded by the Parkland Agriculture Research Initiative (PARI) and Agriculture and Agri-Food Canada, for which I am very grateful.

Table of Contents

Abstract	ii
Acknowledgments	iii
Table of Contents	iv
Chapter 1 Introduction	1
1.1 History	1
1.2 Introduction to Farm Smart 2000	2
1.2.1 Technical Details	4
1.3 The Author's Project Responsibilities and Contributions	5
1.3.1 Objectives of Research Project	6
1.3.2 Knowledge and Data Acquisition	6
1.3.3 Software Architect and Designer of Farm Smart 2000	7
1.3.3.1 The GrowIT Prototype	8
1.3.3.2 PARI DSS '95 Prototype	8
1.3.3.3 PARI DSS '97 Prototype	11
1.3.4 Summary	13
1.4 Overview of Dissertation	15
Chapter 2 Overview of Conservation Farming Systems	18
2.1 Background	18
2.1.1 Changes in Farming Systems	19
2.1.2 Complexity of Conservation Farming Systems	22
2.1.2.1 Land Base	23
2.1.2.2 Crop Rotations	23
2.1.2.2.1 Crop Selection and Water Interaction	24
2.1.2.3 Pest Control	25
2.1.2.3.1 Disease Control	26
2.1.2.3.2 Weed Control	28
2.1.2.3.3 Insect Control	35
2.1.2.4 Residue Management	36
2.1.2.5 Seeding and Fertility	40
2.1.2.5.1 Fertility	43
2.1.2.6 Equipment and Economics	44

2.1.2.6.1	Equipment	44
2.1.2.6.2	Economics	46
Chapter 3	Technology Transfer	51
3.1	Sources of Decision Support	51
3.2	Technology Adoption	53
3.2.1	Categories of Technology Adopters	53
3.2.2	The Adoption Process	54
3.2.2.1	Acquisition of Knowledge of Innovation	54
3.2.2.2	Attitudes and Persuasion Toward an Innovation	55
3.2.2.3	Adoption Decision	55
3.2.2.4	Confirmation of Decision Made	55
3.2.2.5	"Trickle Down" Adoption Process	56
3.2.3	Characteristics of Technology Adoption	57
3.2.3.1	Information and Knowledge Sources	57
3.2.3.2	Personal Characteristics	57
3.2.3.3	Farm Characteristics	58
3.2.3.4	Farm Structure Factors	58
3.2.4	Economics	59
3.2.5	Farm Support Programs	59
3.2.6	Characteristics that Influence Technology Adoption	60
3.2.7	Management of Adopted Technologies	60
3.2.8	Internal Management	62
3.2.9	External Management	62
3.2.10	Behavioral Influences of Technology Adoption	63
3.2.10.1	Psychological Influence	63
3.2.10.2	Profitability Influence	64
3.2.10.3	Farming Life Influence	64
Chapter 4	Problem-Solving and Database Technologies for Decision Support Systems	66
4.1	Problem-Solving Systems	66
4.1.1	Terminology	67
4.1.2	Cooperating Expert Systems/Agents	67
4.1.3	Distributed Artificial Intelligence	69
4.1.4	Multi-Agent Systems	69

4.1.4.1	Methodologies for Multi-Agent Cooperative Problem Solving	70
4.1.4.2	Collaboration and Coordination Among Multiple Agents . .	71
4.1.4.3	Knowledge Requirements for Multi-Agent Collaboration . .	72
4.1.5	Problem of Conflict and Resolution	73
4.1.5.1	Solutions for Conflict Management	75
4.2	Database Technologies for Decision Support Systems	80
4.2.1	Conventional Database Management Systems	81
4.2.2	Distributed Database Systems	82
4.2.3	Heterogeneous/Multi-Databases	82
4.2.4	Database and Knowledge-Base Systems	83
4.2.5	Blackboard Model	85
4.2.5.1	Independence of Expertise	88
4.2.5.2	Diversity in Problem Solving Techniques	89
4.2.5.3	Flexible Representation of Blackboard Information	89
4.2.5.4	Common Interaction Language	89
4.2.5.5	Event-Based Activation	89
4.2.5.6	Need for Control	90
4.2.5.7	Incremental Solution Generation	90
4.2.5.8	Summary	90
4.2.6	Blackboard-Based Databases	91
4.2.6.1	The Blackboard Control Shell	95
4.2.6.2	Legacy Data on the Blackboard	96
4.2.7	Active Databases	97
4.2.7.1	Object-Oriented Modeling Technology	99
4.2.7.2	Rule Management	102
4.3	Conclusion	106
Chapter 5 The Development of Farm Smart 2000		108
5.1	Overview of Farm Smart 2000's Three Tiers of Decision Support	108
5.1.1	Basic Decision Support	110
5.1.2	Advanced Decision Support	112
5.1.3	Interrelated Decision Support	113
5.2	Development of Farm Smart 2000	115
5.2.1	Providing Cooperative Multi-Agent Problem Solving	117
5.2.1.1	Managing Agents	119

5.2.1.2	Distributed Versus Multi-Agent Systems	120
5.2.1.3	Achieving Processing Control	120
5.2.1.4	The Global Control Expert	121
5.2.1.5	Object-Oriented Technology	123
5.2.2	The Development of an Open System Architecture	124
5.2.2.1	The Blackboard Model's Contributions	126
5.2.3	Providing Decision Support for Specific Individual Farming Systems	128
5.3	Development of an Open and Cooperative Problem-Solving System	131
5.3.1	The Blackboard Model/Database	132
5.3.1.1	Data Integrity	133
5.3.1.2	Information Efficiency	134
5.3.2	Methods of Integrating Rules and Active Databases	135
5.3.2.1	An Example Scenario Using Active Database Technology	137
5.4	Data Mining/Discovery Systems	145
	Chapter 6 The Use and Design of Farm Smart 2000	147
6.1	Testing and Validating Farm Smart 2000	147
6.2	Farm Smart 2000 as a Technology Transfer Tool	149
6.3	SmartRKS: Smart Record Keeping System	150
6.3.1	Using SmartRKS	151
6.3.1.1	Toolbars	153
6.3.1.2	Generating SmartRKS Reports	163
6.3.1.2.1	The Report Module	165
6.3.2	Time Ranges	165
6.3.3	Data Validation	166
6.3.4	SmartRKS Technical Details	166
6.4	Farm Smart 2000 Tier-3	167
6.4.1	Using Farm Smart 2000 Tier-3	168
6.4.1.1	Buttons	169
6.4.1.1.1	Add or Modify a Location	169
6.4.1.1.2	Add or Modify a Crop	171
6.4.1.1.3	Function of Other Buttons	172
6.4.1.2	Work-Ahead Problem	174
6.4.1.3	Using the Selection Grid	175
6.4.1.4	Scenarios	176

6.4.1.5	Input Folders	176
6.4.1.5.1	Goals Folder	177
6.4.1.5.2	Tillage Folder	177
6.4.1.5.3	Weed Control Folder	178
6.4.1.5.4	Disease Control Folder	178
6.4.1.5.5	Insect Control Folder	179
6.4.1.5.6	Fertility Folder	180
6.4.1.5.7	Weather Folder	181
6.4.1.5.8	Economics Folder	181
6.4.1.6	Result Folders	182
6.4.1.6.1	Result Grouping	182
6.4.1.6.2	General Results Folder	183
6.4.1.6.3	Crop Production Results Folder	183
6.4.1.6.4	Economic Results Folder	184
6.4.1.6.5	Conservation Results Folder	185
6.4.1.6.6	Crop Protection Results Folder	186
6.4.1.7	Printing Reports	187
6.4.2	Farm Smart 2000 Tier-3 System Components	187
6.4.2.1	Description of Components	189
6.4.2.2	Rules	190
6.4.2.3	The Analysis/Evaluation Process	191
6.4.3	General User Requirements/Specifications	191
6.4.3.1	Analyzing Multiple Crop and Location Combinations	193
6.4.3.2	Results Presentation	195
6.4.3.3	Missing Data	195
6.4.3.4	Managing Management Units	196
6.4.3.5	Delivery of User Explanations of Agronomic Reasoning	196
6.4.3.6	Data Integration	197
6.4.3.6.1	Problem Background	197
6.4.3.6.2	Problem Scenario	198
6.4.3.6.3	Redesign Possibilities	200
6.4.3.6.4	Problem Analysis	201
6.4.3.6.5	Possible Solutions	202
6.4.3.6.6	The Solution Implemented	203

6.4.3.7	Rule Management Conventions	204
6.4.3.8	Installer	205
6.4.3.9	Agent Development Guidelines	206
6.4.3.9.1	Agent Communication with SmartRKS	206
6.4.3.9.2	Agent Communication within Farm Smart 2000	207
6.4.3.9.2.1	Interactive and Non-Interactive Agents	207
6.4.3.9.2.2	Dynamic Data Exchange	208
6.4.3.9.2.3	Dynamic Link Library	209
6.4.3.10	Agents and Knowledge-Bases	209
6.4.3.10.1	Crop Variety Select	210
6.4.3.10.2	Crop Rotation Planner	210
6.4.3.10.3	Crop Protection Planner	211
6.4.3.10.4	Weed Management Planner	214
6.4.3.10.5	Crop Yield Model	214
6.4.4	Other Similar Multi-Agent Problem-Solving Systems	215
Chapter 7	Conclusions	216
7.1	The Achievement of Farm Smart 2000's Goals and Objectives	216
7.2	Technology Transfer and Adoption	217
7.3	Contributions of Research	218
Chapter 8	Further Opportunities for Research and Development	223
8.1	Summarize Results	223
8.2	Support for Alternate and Specialty Crops	223
8.3	Support for Precision and/or Landscape Farming	223
8.4	Agent Framework	225
8.5	Support for Object Linking and Embedding Automation	225
8.6	Support for Graphical Icon User Interface	226
8.7	Free-Form Data Entry	226
8.8	SmartRKS Compatibility and Import/Export Capabilities	226
References		227
Appendices		243
APPENDIX A		243
APPENDIX B		246

List of Tables

Table 2.1	1991 Farm Statistics for Alberta, Saskatchewan, and Manitoba	20
Table 2.2	Acres Affected by Soil Conservation Practice (1985-1989)	20
Table 5.1	Map of Methods/Techniques Developed and Implemented Compared to Alternatives	140

List of Figures

Figure 1.1	Control Interface Providing Access to the Three Tiers of Decision Support	3
Figure 1.2	Main Screen of PARI DSS '95 Prototype	9
Figure 1.3	PARI DSS '95 Prototype System Architecture	10
Figure 1.4	Tier-3 of the PARI DSS '97 Prototype	12
Figure 1.5	A Graphical Representation of the Author's Contributions	14
Figure 2.1	Conservation Farming System Representation	21
Figure 3.1	Characteristics that Influence Technology Adoption	61
Figure 4.2	Blackboard-Based Problem-Solving Concept	86
Figure 5.1	Farm Smart 2000's Three Tiers of Decision Support	109
Figure 5.2	Farm Smart 2000 Tier-3 System Architecture	115
Figure 5.3	Farm Smart 2000 Components	116
Figure 6.1	Scope Selector Showing Displayed Year and "Wheat Fields" Being Expanded	152
Figure 6.2	Land Definitions and Examples	152
Figure 6.3	Scope Selector Showing Pop-up Dialogue	153
Figure 6.4	Access to SmartRKS Database	167
Figure 6.5	Tabs in the Input Folders Window	176
Figure 6.6	Result Tabs	182
Figure 6.7	Tier-3 System Components	188
Figure 6.8	The Analysis/Evaluation Process	192
Figure B.1	Farm Field Definition Screen	248
Figure B.2	Equipment Editing Screen	249
Figure B.3	Events Entry and Editing Screen	249
Figure B.4	Example of Event Report from RKS Module	250
Figure B.5	Input/Results Structure of the Crop Planning Module	251
Figure B.6	Field and Crop Selection Grid for the Crop Planning Module	252
Figure B.7	Input Screen Tabs for the Crop Planning Module	252
Figure B.8	Crop Planning Module Results Structure	253
Figure B.9	Crop Planning Module General Results Folder	254
Figure B.10	Crop Planning Module Crop Production Folder	254
Figure B.11	Crop Planning Module Economics Folder	255

Figure B.12 Cost Comparison Graphs in Crop Planning Module	255
Figure B.13 Crop Planning Module, Conservation Results Folder	256
Figure B.14 Crop Planning Module, Herbicide Details Window	257

Chapter 1: Introduction

This chapter provides a historical perspective leading up to the commencement of the actual research project and an introduction to the research project itself. It also acts as a road map of where to find related information in this dissertation. In the historical perspective, sufficient information is conveyed to provide an appreciation of how and why this research project evolved, and who was involved. The objective of the research project is stated and the resultant decision support system is introduced. The responsibilities and contributions of the author of this dissertation for the research project are discussed. This chapter ends with an overview of the remaining chapters in this dissertation.

1.1 History

The computerized decision support system, called Farm Smart 2000, grew from within a large project called the Parkland Agriculture Research Initiative (PARI) which was proposed by Agriculture and Agri-Food Canada (AAFC). The PARI's objective was to develop and demonstrate soil and water conservation technology in the Parkland⁴ soils of the prairies. This objective was based on the premise that long term sustainability of the soil resource base is fundamental to agriculture. Excessive tillage and frequent summerfallow occurs in the Black and Gray soil zones of the prairies even though moisture conditions are usually adequate for continuous crop production. Excessive tillage and summerfallow results in loss of crop residues, depletion of soil organic matter, release of greenhouse gases to the atmosphere, increased potential for soil erosion, and increased rates of general soil degradation.

The PARI consisted of three main programs:

- 1) Research and refinement of sustainable management systems.
- 2) Soil resource monitoring and evaluation.
- 3) Landscape-scale farming research and technology transfer.

⁴ The Parkland region consists of the Black and Gray soil zones, and Aspen Parkland Ecoregion of the Prairie Provinces.

These programs increased the knowledge base and access to information on soil conservation practices and cropping systems on the Prairies. The entire PARI was a collaborative venture involving mainly Western Canadian specialists from government, university, private industry and, most importantly, producers (i.e. the end-users) from various farming communities. The PARI was supported by Alberta Agriculture, a provincial government department, who participated in the management of the initiative.

A sub-program within the first main program was to develop one or more expert systems to deliver research information and other expertise on conservation farming systems to extension personnel and producers. It was proposed that the development of an expert system may provide the link between researchers and extension personnel and producers to transfer technology, information, and knowledge on methods to reduce summerfallow, crop rotations to replace summerfallow, and the economic and environmental advantages that result from the minimization of summerfallow (i.e. conservation and sustainable farming systems).

1.2 Introduction to Farm Smart 2000

Farm Smart 2000 as described in Section 5.1 and illustrated in Figure 5.1 provides decision support to end-users utilizing three tiers that are accessible via the control interface as illustrated in Figures 1.1 and 5.3.

Tier-1, described in detail in Section 5.1.1, is called the "information" level because it provides information or Basic Decision Support via a web site on the Internet. Farm Smart 2000's web site contains a wealth of agronomic information which is disseminated in different formats including factsheets, newsletters, informational bulletins, answers to frequently asked questions, and a specialist directory called Who's Who⁵. With Tier-1 being an Internet web site, it resides on the PARI Decision Support System (DSS) server (<http://paridss.usask.ca>). As mentioned, Farm Smart 2000 is distributed on CD-ROM, but when the user requests access to tier-1 via the Farm Smart 2000 control interface, the user's Internet web browser is launched with the "location" set to the Farm Smart 2000 home page.

⁵ The Who's Who specialist directory is implemented but is not currently populated. The other formats exist on the web site and will be more widely utilized as the development of the Farm Smart 2000 web site and its sub-sites continue.



Figure 1.1 Control Interface Providing Access to the Three Tiers of Decision Support

Early in 1994 before standardized world-wide-web browsers (e.g. Netscape Navigator and Internet Explorer) were available, the PARI DSS had an Internet management information system, based on Gopher technology, to provide a document search and retrieval system for distribution of newsletters, bulletins, and other information regarding conservation farming. However, with the rapid growth of Internet capabilities, this PARI DSS Gopher was short lived becoming first the PARI DSS web site and then the Farm Smart 2000 web site as Farm Smart 2000 developed.

Tier-2, described in detail in Section 5.1.2, is called Advanced Decision Support and is a collection of stand-alone expert systems, each focused at providing decision support for specific narrowly defined problems and are contained in a software "tool box". This software "tool box" contains the expert systems that were created internally and externally by collaborators. These stand-alone expert systems in tier-2 are distributed on

the Farm Smart 2000 CD-ROM. To manage distribution, copyrights, and intellectual property rights, these expert systems are not available for downloading via the Farm Smart 2000 web site, although software updates will be available via the web site (e.g. paridss.usask.ca/crop_planner/downloads.html).

Tier-3, the top most level, is called Interrelated Decision Support. In tier-3 Farm Smart 2000 integrates multiple knowledge sources (e.g. expert systems and knowledge-bases) into one comprehensive decision support component. The Interrelated Decision Support tier is actually called “Crop Planning” in the Farm Smart 2000 system (See Figure 1.1) to better represent its function to the producer. Tier-3 is also distributed on the Farm Smart 2000 CD-ROM with data updates eventually being downloadable from the Farm Smart 2000 web site.

A customized record keeping system called SmartRKS was developed specifically for use with Farm Smart 2000 (See Figure 5.1 and Section 6.3), as a result of failing to find a suitable existing record keeping system to utilize. SmartRKS is capable of storing and retrieving on-farm data, performing calculations and summations, and reporting results. SmartRKS is only accessible by tier-3 via the Global Control Expert although plans for future development include tier-2 having access to the SmartRKS (See Figure 5.3 and Section 8.8). SmartRKS is a stand-alone application and has been distributed on CD-ROM and tested as such, although plans are to ultimately distribute it solely with Farm Smart 2000 as a total software package. Hence, the Farm Smart 2000 CD-ROM contains the web browser launch application (i.e. tier-1) as described above, tier-2, tier-3, and SmartRKS.

1.2.1 Technical Details

Farm Smart 2000 is distributed on a CD-ROM or 13 diskettes with computer system requirements being:

- Windows 95, Windows 98, Windows NT 3.51, or Windows NT 4.0 operating system
- 486 or greater, PC Compatible computer (Pentium computer or faster recommended).
- 16 MB of RAM, minimum (32 MB recommended).
- 30 MB of Free Disk Space.

Farm Smart 2000 was developed for 32-bit *Microsoft Windows*[™] using *Microsoft Visual C++*[™] version 5.0. It requires run-time versions/licenses of Neuron Data's *Elements Environment*[™], specifically *Open Interface Element*[™], which is now called *Blaze Presenter*[™], and *Intelligent Rules Element*[™], which is now called *Blaze Expert*[™]. These run-time versions/licenses are an integral part of the Farm Smart 2000 software application and distribution (See Section 1.3.3.3), with the user being unaware of their presence.

When SmartRKS is used as a stand-alone record keeping system it is distributed on CD-ROM or 5 diskettes with system requirements being:

- Windows 95, Windows 98, Windows NT 3.51, or Windows NT 4.0 operating system;
- A 486 or greater, PC Compatible computer (Pentium computer or faster recommended) computer system;
- 16 MB of RAM, minimum;
- 15 MB of free disk space.

1.3 The Author's Project Responsibilities and Contributions

The author, as Project Manager of PARI DSS, proposed at the onset of the project in the fall of 1991 that a decision support system, integrating several expert systems, databases, and knowledge bases, would better fulfill the requirements of the project for the following reasons:

- 1) Although expert systems would provide an excellent capability to transfer technology, knowledge, and information to extension specialists and producers, expert systems should remain narrow in breadth as they are meant to be, in order to be successful [Barrett and Jones, 1989], focusing on mimicking the expertise of one or more experts in only one specific area of knowledge (e.g. fertility, weeds, economics, organic residue, rotations, or varieties). Furthermore, the complexity and broad-based problems that can be solved by several expert systems (or expert humans) working together, far exceeds the problem-solving capabilities that could be obtained by developing a single expert system.

- 2) With agricultural technology continually advancing, it was important to build a system with an open architecture that can be readily updated, expanded, and enhanced as new knowledge (e.g. additional expert systems) emerge as a result of laboratory and field plot research and development, without re-engineering the entire system.
- 3) With the PARI DSS initiative's inherent collaboration⁶, it was beneficial to be able to provide freedom to collaborators to transform their own specific areas of expertise into expert systems (assuming they had the resources to do so), providing them with their own useful decision tool and a sense of ownership, while enabling the PARI DSS to capitalize on this expertise through the integration of these expert systems.

1.3.1 Objectives of Research Project

The author's proposal was accepted and the objective of this research project was established to develop the necessary computer system architecture to initially integrate the knowledge and expertise of several human agricultural experts, allowing for future expansion and enhancements (i.e. an open system), and to simulate these experts working cooperatively to solve broad-based agricultural problems (e.g. crop rotation planning). The following sub-sections describe specifically, in terms of this dissertation, what the author was responsible for and contributed.

1.3.2 Knowledge and Data Acquisition

The author collaborated and cooperated with human experts⁷ seeking and acquiring their knowledge for the creation of expert systems, rules, and/or knowledge bases⁸. A Steering Committee was established consisting of twelve members (i.e. liaisons) who provided suggestions and advice to the author from within their respective organizations. This Steering Committee included three producers as members. Three working groups were established with three members in each, one in Lethbridge Alberta, one in Edmonton Alberta, and one in Saskatoon Saskatchewan. The author oversaw these working groups

⁶ The author presented 87 formal presentations to potential collaborators at conferences, seminars, workshops and meetings during the course of the research project.

⁷ The author worked closely with 47 primary experts plus 121 secondary experts.

⁸ The methods utilized for knowledge acquisition/elicitation, which are part of the expert system development methodology, are beyond the scope of this dissertation and thus not discussed.

and was the Working Group Leader of the Saskatoon group. These working groups acted as a “sounding board” for the author’s ideas and technical designs. In addition, the working groups collected agronomic data, evaluated software⁹ including components and prototypes of Farm Smart 2000 as well as internally developed expert systems for correctness, and external expert systems and models for possible usefulness (e.g. Lands)¹⁰. Every Steering Committee meeting and Working Group meeting was a challenge to negotiate consensus on controversial issues and direction. Appendix A is an example which illustrates the differing opinions among the PARI DSS Steering Committee members when asked what they thought were the ten most important producer issues for conservation cropping decisions.

In addition to knowledge, there was naturally a requirement for agronomic data for use in SmartRKS, tier-3, and by the stand-alone expert systems, for which the author was responsible. Hence, early in the PARI DSS project several agronomic datasets were developed from data collected across the Prairies. Some of this data collection was contracted out and the Lethbridge Working Group was also heavily involved. Data dictionaries can be accessed on the Farm Smart 2000 web site at http://paridss.usask.ca/research_datasets.html for many of the datasets/databases which are not protected by intellectual property rights.

1.3.3 Software Architect and Designer of Farm Smart 2000

The author was the overall architect and designer of the resultant Farm Smart 2000 system. This included developing the specifications and methodologies for Farm Smart 2000, but not including the agents, except one, the Weed Management Planner (See Sections 5.1.2 and 6.4.3.10.4). The Weed Management Planner was designed by the author with cooperation from several weed experts for knowledge acquisition. The Weed Management Planner is a sophisticated expert system available within Farm Smart 2000.

⁹ The software quality metrics used, which are a part of the project management process, are also beyond the scope of this dissertation and thus not discussed.

¹⁰ A Land Analysis and Decision Support System called LANDS was thought to possibly be an answer for the PARI DSS. After thorough system evaluation by the author, including the development of a “proof of concept”, which LANDS did not achieve, the author concluded that LANDS, being a turn-key system, did not have sufficient flexibility to be adaptable to fulfilling the requirements of the PARI DSS.

The following sub-sections describe the prototypes that were designed and developed, leading up to the resultant Farm Smart 2000 system.

1.3.3.1 The GrowIT Prototype

After proposing that the PARI DSS should be a decision support system consisting of multiple expert systems integrated to work together, the author designed and implemented a dBase IV relational database prototype to prove that two commercial agriculture packages, specifically the Crop Rotation Planner and the Herbicide Planner could be integrated to work together.

The GrowIT prototype¹¹ was the resulting application which was centered around a relational database consisting of two sets of relations. The primary set of relations supported the acquisition, long term storage, and retrieval of data as they pertain to crops and weed control by herbicides. The Crop Rotation Planner and Herbicide Planner accessed the relational database in order to move data between the two planners during the problem-solving process, as well as to update or insert data into the database for storage and retrieval. A secondary set of relations acted as a data dictionary to translate between different dialects of names which are used for crops, weeds and herbicides. For example, the crop name for peas can be "field peas" or just "peas". This data dictionary translated names into a standard and supported the coexistence of the Crop Rotation Planner and the Herbicide Planner. These different dialects of names have been a concern throughout the development of Farm Smart 2000 (See Section 8.7)

Although GrowIT was successful in integrating two expert systems to work together, it had only limited problem-solving capabilities. The integration was awkward, limiting the extent of communication between the expert systems, and was not conducive to the development of an open system.

1.3.3.2 PARI DSS '95 Prototype

After GrowIT, the next prototype was called PARI DSS '95, a prototype that was developed and demonstrated at the PARI '95 Workshop entitled "Bringing Conservation Technology to the Farm" and held at the Delta Bessborough in Saskatoon November 26-28, 1995 (See Figure 1.2). The PARI DSS '95 system specifications¹² and architecture

¹¹ The GrowIT System Specifications document was 18 pages in length.

¹² The PARI DSS '95 System Specifications document was 47 pages in length.

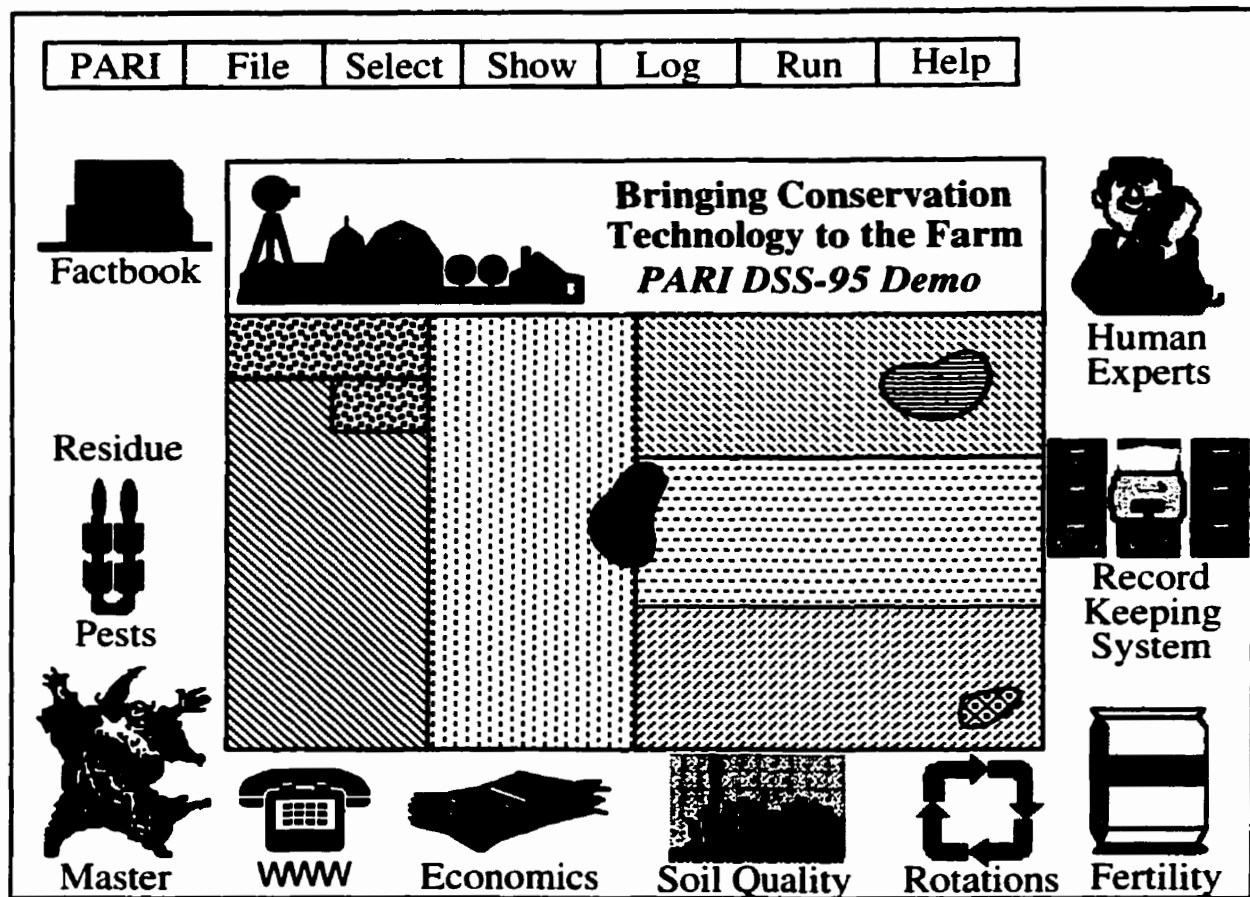


Figure 1.2 Main Screen of PARI DSS '95 Prototype

were more complex than GrowIT, providing more functionality and more “decision support” capabilities (See Figure 1.3). Although PARI DSS ‘95 did not have the control interface of Farm Smart 2000, it essentially did have access to the same 3 tiers of decision support via an user interface. This user interface provided access to the PARI DSS web site, several stand-alone expert systems, and a “Master” which was the user interface for tier-3 (See Figure 1.3).

As illustrated in Figure 1.3 four heterogeneous reusable agents (i.e. expert systems) were integrated in tier-3 including the original two expert systems from GrowIT; the Crop Rotation Planner and the Herbicide Planner. The Global Control Expert in PARI DSS '95 provided the global control necessary for interaction, communication and coordination among agents. The Global Control Expert in the PARI DSS '95 prototype consisted mainly of problem-solving processes to answer specific questions. These problem-solving processes could be altered during the course of control by constraint and goal rule sets.

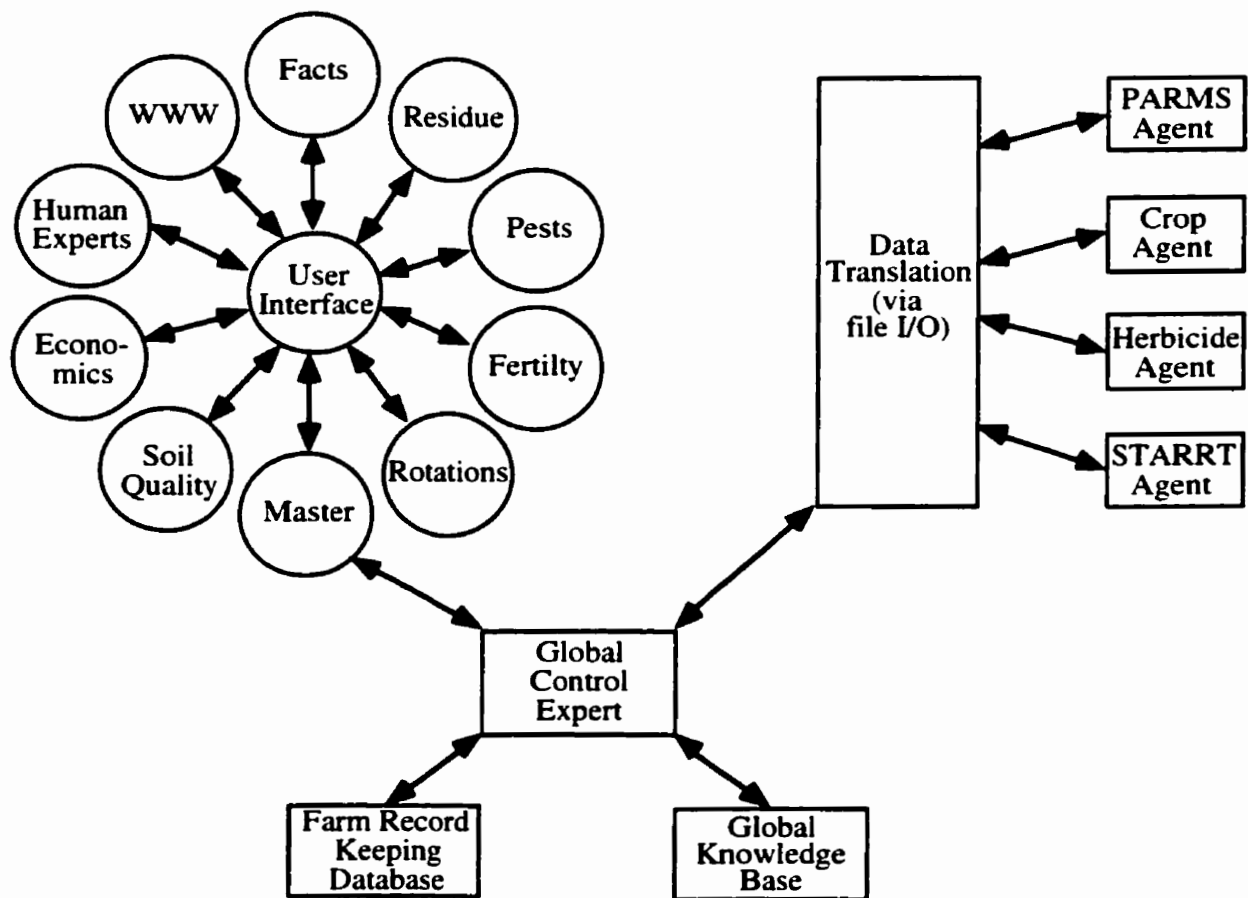


Figure 1.3 PARI DSS '95 Prototype System Architecture

The Global Knowledge Base was a data structure which contained various knowledge, including problem-solving techniques, solution evaluation criteria, minimum and maximum constraints, conflict threshold limits, and other relevant meta-knowledge. The Global Control Expert had access to meta-knowledge about the reusable agents, which was contained within the Global Knowledge Base, and was used to solve inconsistency issues among the agents. The Global Control Expert triggered agents into action based on influence from the agents' contributions, the knowledge of other agents, and the state of the problem-solving process. Since there is no common communication language shared among the agents, the Global Control Expert monitored the agents' processing of tasks, utilizing the Global Knowledge Base, providing the necessary interaction and communication among the agents.

Minimum and maximum constraints, goal rule sets, and conflict threshold limits were the basis for resolving conflict which governed system coordination. In the PARI

DSS '95 prototype, the user selected crop production questions from a limited list of only three, which best matched his/her particular problem and then the system solved the problem (i.e. answered the question) based on user input. These questions included the following:

- 1) Which crop(s) could I grow?
- 2) Which field should I grow this crop on?
- 3) What is likely to happen with this crop grown on this field?

Although PARI DSS '95 utilized successful methodology, it was quite limited, in that it could not adequately represent the complex interrelations of conservation farming systems resulting in provision of only general decision support for specific problems/questions. One of the most limiting factors was that there was no global area (i.e. blackboard) to which agent contributions and intermediate problem-solving states and information could be posted making the Global Control Expert solely responsible for this and its other "control" tasks, frequently resulting in poor performance.

1.3.3.3 PARI DSS '97 Prototype

The PARI DSS '97 prototype was developed to demonstrate to AAFC's Western Region Director General and Western Region Directors that the goals of the PARI DSS had been achieved. These senior executives agreed unanimously that the PARI DSS goals had been met and that they would continue to finance further development of PARI DSS '97 from their own Research Centres' funds since the 5-year PARI had concluded officially. The PARI DSS project was the only program from the PARI to receive support of this kind beyond the PARI's 5-year time frame.

The PARI DSS '97 prototype¹³ contained similar methodologies and techniques as those contained in Farm Smart 2000 with the most important addition, not present in the PARI DSS '95 prototype, being the design of a blackboard database, consisting of a global data area through which the agents could communicate and interact (See Figure 1.4 compared to Figure 5.3). Rather than the Global Control Expert, as in the PARI DSS '95 prototype, attempting to manage the intermediate problem solutions and problem-solving

¹³ The PARI DSS '97 System Specifications document was 84 pages in length.

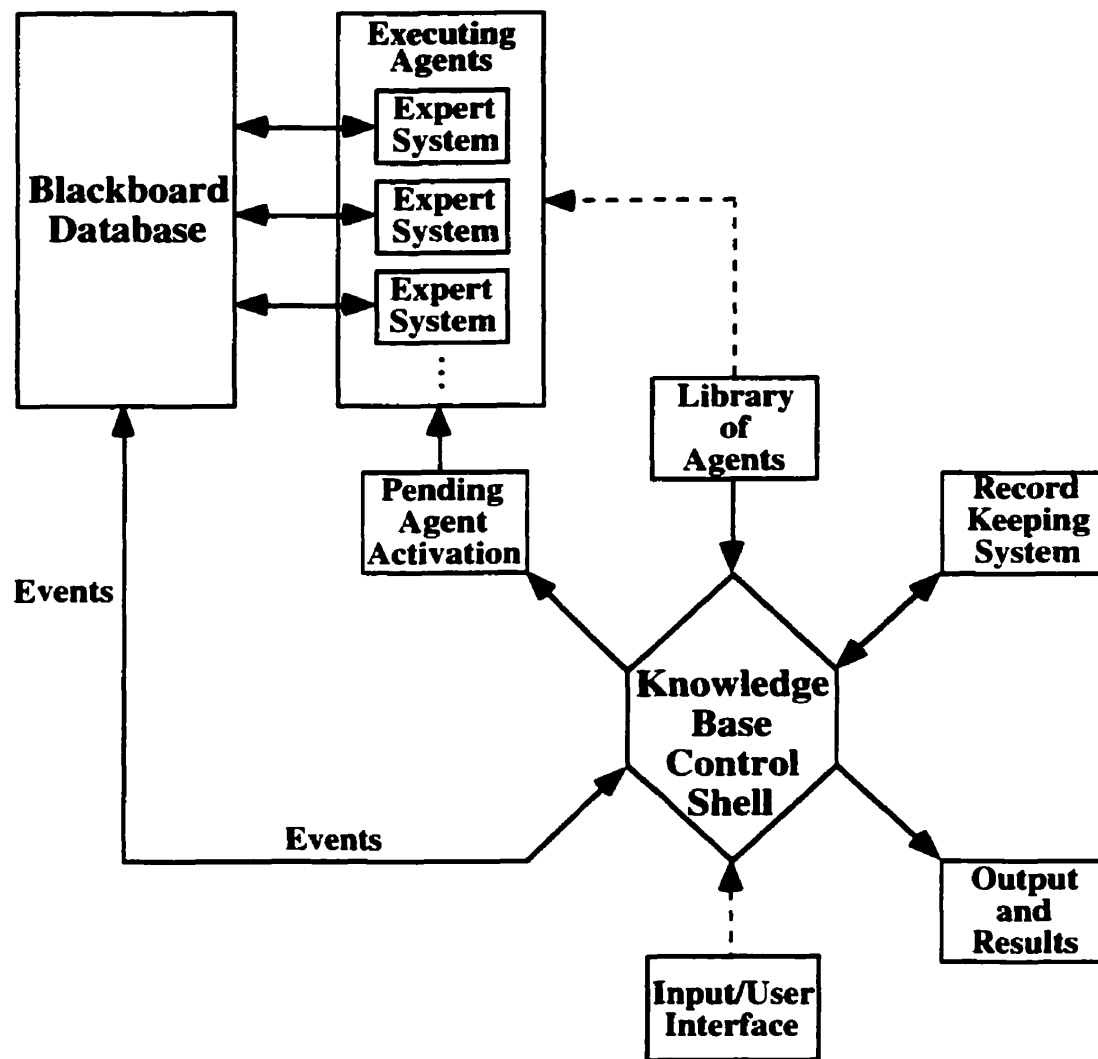


Figure 1.4 Tier-3 of the PARI DSS '97 Prototype

state data, agents in the PARI DSS '97 prototype incrementally made changes to the blackboard based on their problem-solving ability at any given time.

With the blackboard database being an integral part of PARI DSS '97 prototype, and later Farm Smart 2000, it was critical that the right software application tools be selected by the author in order to build an open architecture for multi-agent cooperative problem solving. After a thorough investigation, three blackboard frameworks were researched in depth. The *BB1*TM System by Barbara Hayes-Roth from Stanford University and the *GBB*TM (Generic Blackboard Builder) framework, originally from University of Massachusetts at Amherst, and now commercially available from Blackboard Technology Group, Inc., were similar in functionality and capability. Both of these were

developed to support a “blackboard application” based on particular blackboard methodology, rather than supporting the more general “blackboard model” (See end of Section 4.2.5) The third software was from Neuron Data’s (now Blaze Software) *Elements Environment*[™] of which two components (i.e. elements) were considered for use in PARI DSS ‘97. These elements are the *Intelligent Rules Element*[™] (now called *Blaze Expert*[™]) and the *Open Interface Element*[™] (now called *Blaze Presenter*[™]) (See Section 1.2.1). Whereas the control shell of *BB1*[™] and *GBB*[™] were confining, limiting control flexibility, the *Intelligent Rules Element*[™] provided the required tools to build a custom Global Control Expert providing capabilities that conform to Farm Smart 2000’s goal-directed and data-directed control requirements as described in Section 5.2.1.3. This was accomplished even though the *Intelligent Rules Element*[™] is advertised as a solution for building “business rules” in intelligent applications. The *Open Interface Element*[™] is a tool for automating the development of Graphical User Interfaces (i.e. GUIs) and was/is used to implement the user interfaces in PARI DSS ‘97, and subsequently Farm Smart 2000.

The author’s decision to utilize *Elements Environment*[™] Software was supported by contacts from the Canada Space Agency in Ottawa, which were made at a United States Department of Agriculture Workshop in Beltsville. One software engineer from this agency concluded that if they had the opportunity to rebuild their blackboard application, they would use Neuron Data’s *Elements Environment*[™] Software rather than Blackboard Technology’s *GBB*[™].

1.3.4 Summary

The author was the overall architect and designer of Farm Smart 2000 including the initial prototypes. Most of the programming was contracted out, except for GrowIT, which the author programmed. With PARI DSS being a collaborative project, other partners contributed knowledge and information to the project as well. The shaded area in Figure 1.5 graphically represents the author’s contributions.

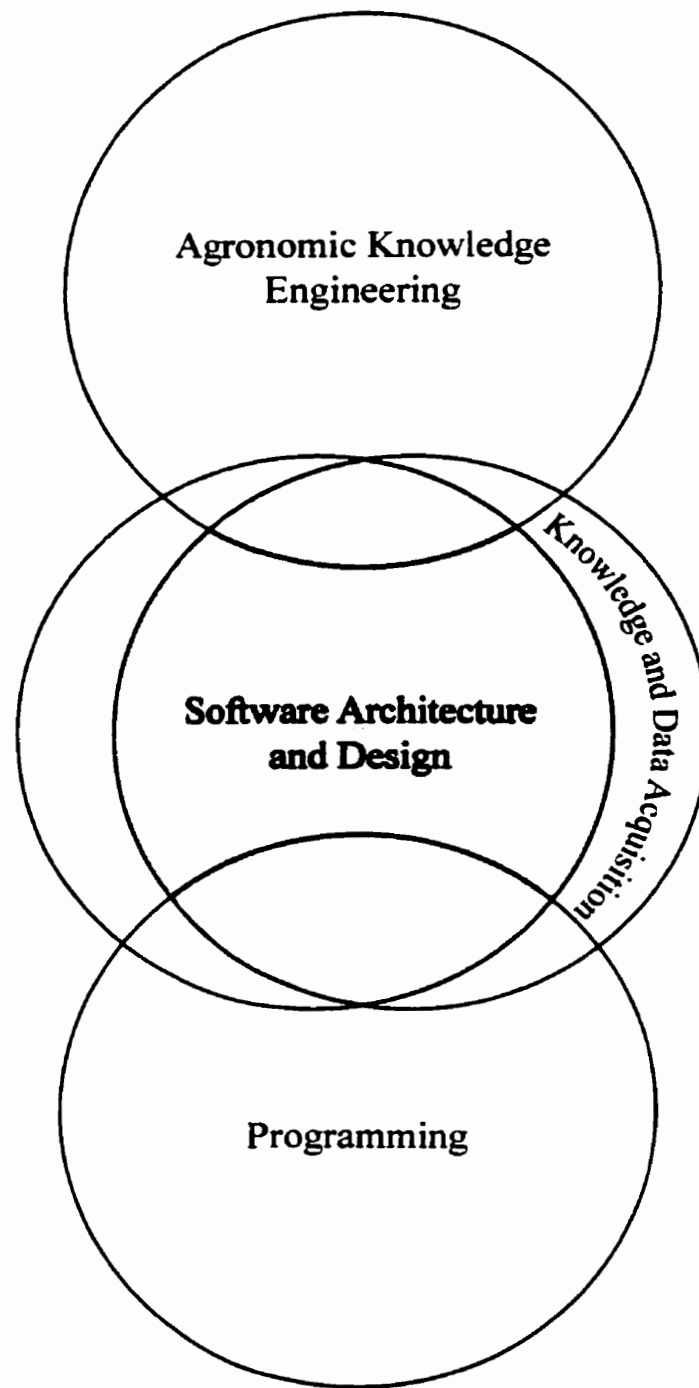


Figure 1.5 A Graphical Representation of the Author's Contributions (Shaded Area)

1.4 Overview of Dissertation

Chapter 1 has provided historical background, discussing how the research project began, who was involved, its goals and objectives, and contributions of the author. It included a description of the prototypes prior to Farm Smart 2000.

Chapter 2 provides an overview of conservation farming systems and introduces the challenges which face agricultural crop production producers. It explains the concerns of soil degradation and sustainable agriculture and the producers' shift to conservation and diversified cropping farming systems to reduce summerfallow and monoculture cereal production. It also explains the complexity of conservation farming/tillage systems as a result of complex interrelationships that exist among farming entities. Furthermore, Chapter 2 describes, that in order to manage the many complex issues of conservation and diversified cropping farming systems and remain economically viable, farm managers need access to information, knowledge and multiple experts that are specific to their farming operation.

Developing a decision support system to aid producers in their decision making is beneficial only if it is adopted. Chapter 3 discusses the results of the literature review, describing technology adoption in terms of the different types of technology adopters, the technology adoption process, the characteristics of technology adoption, economics, farm support programs, characteristics that influence technology adoption, management of adopted technologies, and behavioral influences of technology adoption. Chapter 4 also discusses the results of the literature search and describes several problem-solving systems including cooperating expert systems and multi-agent systems and explains the problem of conflict and how it can be resolved. Chapter 4 also discusses various database technologies for decision support systems including conventional database management systems, distributed database systems, heterogeneous/multi-databases, database and knowledge-base systems, the blackboard model, and blackboard-based databases, as well as active databases.

Chapter 5 focuses on the development of Farm Smart 2000, a prairie crop production decision support system. After providing an overview of Farm Smart 2000's three tiers of decision support (i.e. Basic Decision Support, Advanced Decision Support

and Interrelated Decision Support), Chapter 5 discusses the provision of decision support for specific individual farming systems. Tier-3 is then addressed with a discussion of the development of an open system architecture within a decision support system, and the methodology required to provide cooperative multi-agent problem solving. Discussion of these latter two methodologies is extended to describe their being joined to produce an open and cooperative problem-solving system based on the blackboard model/database, and methods of integrating rules and active databases.

Chapter 6 discusses the design and use of Farm Smart 2000. This chapter begins by discussing how Farm Smart 2000 was tested and validated and how it is used as a technology transfer tool. It describes how SmartRKS, the integral record keeping system, in combination with Farm Smart 2000, are used; the input data required and the results provided. The system components of Farm Smart 2000's tier-3 are described, as are the general user requirements and specifications of tier-3, with a discussion of problems that arose in fulfilling these requirements, and solutions, compromises, and limitations that resulted in the implementation.

Chapter 7 concludes that a decision support system consisting of three levels was designed, developed, implemented, and tested in order to support the farm manager's decision making process. It discusses indicators that substantiate that Farm Smart 2000's goals and objectives were achieved. It further concludes that it is possible to provide technology transfer in the complex subject of conservation farming using integrated multiple agents (i.e. several expert systems mimicking human experts) in a cooperative problem-solving environment. This chapter ends with a section describing how this research and the development of Farm Smart 2000 has assisted in furthering the science and technology of decision support systems. The section summarizes the major problems other researchers discovered which were solved in the development of Farm Smart 2000. It also mentions recent research work by other authors which might be considered if the development of an entirely new Farm Smart 2000 was undertaken.

Chapter 8 lists and describes further opportunities for research and development of Farm Smart 2000 and difficulties which could be addressed. Some of the discussion pertains to further simplifying data entry and enhancing the presentation of results, the

support for additional crops and precision farming, improving the framework for integration of future agents, support for a graphical user interface, and import/export capabilities with third party accounting and financial applications.

Chapter 2: Overview of Conservation Farming Systems

This chapter discusses changes in farming systems and describes the complexity of conservation farming in terms of land base, crop rotations, pest control, residue management, seeding and fertility, and equipment and economics.

2.1 Background

Soil degradation and sustainable agriculture combined with increasing costs of farm inputs and grain transportation are key concerns on the Canadian Prairies and are causing producers to closely examine their land use practices. Producers are beginning to extend and diversify their crop rotations resulting in less summerfallow and monoculture cereal production. Conservation production systems utilizing minimum- and zero- tillage management practices, integrated pest management, and "bottom-line farming" are being implemented to not only sustain agriculture but ensure it is economically viable. Conventional farming practices are not without complications, but today's conservation farming systems are very complex [Ikerd, 1991].

The complexity of conservation farming systems is a result of the complicated interrelationships that exist among farming entities. For example, seeding operations with weed management, crop rotation with plant disease, and residue management with direct seeding. In this research, the knowledge of these interrelationships were acquired from credible sources including human experts and published material, and developed into an artificial intelligent computer application in the form of a decision support system called Farm Smart 2000, capable of dynamically solving crop production problems.

Conservation farming systems and diversified cropping requires different inputs (i.e. fertilizer, herbicides, water, heat and field management), control of a broader spectrum of crop pests (i.e. weeds, insects and diseases) and concludes with different harvest results depending on growing conditions such as heat and moisture. For example, residue management by itself is a challenge, with different crops producing different amounts of crop residue, with different biodegradable characteristics, which can result in seeding or

soil erosion problems. In order to make appropriate farming decisions, while also maintaining economic viability, producers need sound advice and accurate information on all aspects of their farm operations.

Furthermore, to manage the many complex issues of diversified farming and numerous alternatives available, farm managers need access to multiple experts. Cropping decisions based on inaccurate information, or lack of understanding of the processes and consequences involved, can be detrimental environmentally and economically. Harmoniously joining diverse knowledge from cooperating human experts provides an extremely important source of balance and robustness in many real-world situations. This is characterized by carefully selected human project teams and work groups. Teams and groups can solve problems which are normally beyond the comprehensiveness of individual experts, and in doing so, provide potentially creative and innovative solutions resulting from a rich and varied body of knowledge [Lander and Lesser, 1989a]. Thus, mimicking this human problem-solving methodology by integrating artificially intelligent agents has advantages directly related to the real world. However, developing artificial intelligent cooperative problem-solving systems is not an easy task as stated by Lander and Lesser (1994, p.13); "There is a high degree of complexity inherent in building heterogeneous agents that can understand each other well enough to positively affect mutual work."

2.1.1 Changes in Farming Systems

In an effort to adapt to changing markets and farm conditions, Canadian Prairie producers are adopting new crops and new cropping systems to better compete with a more market driven economy. In the last 25 years, farming on the Canadian Prairies has progressed from what can be considered elementary cropping systems of fallow and cereal crops, to sophisticated and complex conservation farming and direct seeding systems. Throughout the 1970s, 1980s and early 1990s, seeding equipment and agronomic methodologies and practices have been extensively researched and developed to enable conservation farmers to reduce their impact on the soil resource. Table 2.1, analyzed from 1991 Census of Agriculture for the three prairie provinces, shows the growing trend towards less summerfallow.

Table 2.1: 1991 Farm Statistics for Alberta, Saskatchewan, and Manitoba

	Alberta	Saskatchewan	Manitoba	Total
Number of farms	57,245	60,840	25,706	143,791
Total area (acres)	51,425,111	66,386,074	19,088,868	136,900,053
Farms with land in crops	50,732	58,650	23,563	132,945
Land in crops (acres)	22,961,142	33,257,706	11,764,813	67,983,661
Farms with summerfallow	18,963	45,577	7,511	72,051
Summerfallow (acres)	4,377,212	14,116,713	733,899	19,227,824

Institutional and structural changes in soil conservation activities during the late 1980s had a significant impact on adoption of soil conservation practices. Even in a time of widespread agricultural uncertainty, adoption of soil conservation technologies were impressive as illustrated in Table 2.2, and continue to be in the 1990s.

Table 2.2: Acres Affected by Soil Conservation Practice (1985-1989)

	Saskatchewan (acres)	Manitoba (acres)
Forage establishment	14,900	14,300
Extended rotations	18,200	80
Conservation tillage	121,000	29,300
Shelterbelts	51,500	152,000
Barriers and cover	17,100	7,680
Salinity diagnosis	70,100	22,000
Total Acres	292,000	225,000

From: R.J. Wettlaufer and P.B. Brand, "Adoption of Soil Conservation Practices on the Canadian Prairies," Proc., Honolulu, Hawaii, October 1992, pp. 401.

This progression of cropping systems has necessitated increased levels of decision support because of their ever increasing complexity. Changing farming systems can be a knowledge seeking experience, involving many decisions which affect the viability of the farm. Figure 2.1 represents a conservation farming system in which direct seeding is utilized as a means of reducing costs and tillage. In the 1950s, the use of discers and one-ways for seeding and extensive cultivation depleted the soil of most surface residue leaving it susceptible to erosion. In the 1970s, air seeders combined seeding with cultivation but were still full tillage machines leaving little residue after seeding particularly since most seeded fields were preworked. Direct seeding, which is a system where no tillage is used prior to seeding and strives to retain anchored surface residue, emerged in the 1980s. The amount of soil disturbance during direct seeding depends on the type of machine, opener

type, and method of placing fertilizer. Retention of residue is the primary focus in a conservation farming system, as illustrated in Figure 2.1, with its importance being many fold. A zero tillage system employing direct seeding with one pass and low soil disturbance is the closest annual cropping system to the original native grass ecology of the prairies. When a new crop is planted directly into the stubble of the previous crop, the old roots and the new roots of the emerging plants anchor the soil and provide aeration and water channels. This is similar to the cycle of the original native grasses where it too was harvested each year with the remainder of the plant slowly decaying on the surface, recycling nutrients back into the soils for future crops.

2.1.2 Complexity of Conservation Farming Systems

There are many direct seeding systems being employed by prairie farmers in conservation farming systems and all have some limitations. There is no perfect conservation farming system or single recipe for direct seeding but rather farm managers need to determine which system works best in their area for the crops they want to grow, and most importantly, that best achieves their farm's goals.

Figure 2.1 illustrates the major components and complexities involved in a conservation farming system. Although not every detail of conservation farming could be included, Figure 2.1 and the following discussion clearly show the complexity involved due to the large number of interrelated crop production entities.

Conservation tillage refers to any crop production system that optimizes the conservation of soil and water resources by maintaining a protective cover of crop residue on the soil surface. This is applicable to high and low disturbance direct seeding and other systems where the number of tillage operations is reduced. The Saskatchewan Soil Conservation Association (SSCA) defines "low disturbance seeding" to be where less than 30% of the soil surface is disturbed and "high disturbance seeding" to be where more than 30% of the soil surface is disturbed [PAMI, 1999]. Zero tillage refers to low disturbance seeding, minimum tillage to high disturbance seeding, and conventional tillage to full tillage systems. The interest in conservation tillage has grown steadily as illustrated in Table 2.1, and continues even more so in the last decade. Although the potential for lower operating costs and the greater availability of suitable equipment have been major factors influencing

the adoption of conservation tillage, it also offers agronomic and environmental benefits over conventional crop production including [PAMI 1999]:

- moisture conservation
- improved crop yields and economic returns
- reduced erosion potential
- enhanced soil productivity
- reduced greenhouse gas emissions

A conservation farming system does not merely reduce the number of tillage operations by replacing them with herbicide applications, but rather is a system of interrelated crop management practices.

2.1.2.1 Land Base

The land base pictured in the centre of Figure 2.1 is the basis for a conservation farming system. Since the amount of residue and organic carbon are determinants of conservation tillage, continuous cropping systems are inherent in conservation farming systems. However, certain soil types, texture and topography are considered too fragile for annual crop production due to factors such as coarse soil texture, steep slopes, severe salinity, frequent flooding, or an abundance of stones [Domitruk et al., 1997]. These factors may vary between soil zones and/or climatic areas. For instance, sandy loam soil in the Brown soil zone could be considered to be as fragile as coarser, loamy sand in the Black soil zone because Brown soil is more prone to drought and windy conditions causing erosion. Climate is depicted in Figure 2.1 by the amount of sun (i.e. heat units) and precipitation that is normal for an area. To further illustrate the complexity of direct seeding associated with specific soil types, there could be an increase in soil erosion on clay soil types, not having significant residue cover, if the direct seeding involved minimal soil disturbance of dry topsoil. Although this seems illogical, clay soils, initially in a conservation farming system, tend to be very erodible in early spring because soil aggregates have broken down due to the action of wind, rain, and freeze-thaw cycles.

2.1.2.2 Crop Rotations

Crop rotations as shown on the land base in the centre of Figure 2.1 are the basis for a successful conservation farming system. A crop rotation is an at least partially

repetitive sequence of crops grown on a field over a period of years. Crop selection is heavily influenced by market trends and the cost of inputs for specific crops as illustrated in Figure 2.1. Initially, a conservation farming system can appear more costly due to capital costs for new or modified seeding machinery and potentially higher herbicide costs [Evans, 1993]. Canada Thistle, for example, is a difficult weed to control without tillage, but pre-harvest glyphosate has shown to be effective. There is excellent machinery available for conservation farming systems and there will be more with ongoing competition and innovation. A direct seeding system that includes moisture conservation methods, such as tall stubble for snow trapping, will often improve the chance of successful stubble cropping in the drier areas of the prairies. The cereal-fallow rotation is still dominant in the Brown soil zone, but many farmers have moved to longer rotations including lentil, chickpea, cereal and oilseed crops.

The success of a direct seeded cereal crop after a lentil crop is limited by the inability of lentil's short stubble to trap snow. However, snow trapping can be enhanced by seeding other taller crops (e.g. sunflower or wheat) as barrier strips in the lentil field. Since the amount of residue left after a lentil crop is quite small, some of the same problems associated with direct seeding after fallow can also be evident when seeding into lentil stubble.

In the Dark Brown soil zone, canola, mustard, and pea are often included in rotations in addition to lentil. Including crops in the rotation that produce tall stubble and more residue (e.g. canola and mustard) help reduce the soil conservation and snow trapping problems associated with lentil. Incorporating snow trapping techniques are useful as available moisture is the key to success. Methods involving stubble shaping and leaving taller stubble can be less expensive and more effective than snow ridging or plowing. However, leaving taller stubble for snow trapping must be balanced with the ability of seeding implements to clear the tall residue and seed correctly.

2.1.2.2.1 Crop Selection and Water Interaction

The crop-water interaction in Figure 2.1 refers to optimizing snow collection, crop rotations and water use patterns [PAMI, 1999]. In this interaction, historical rainfall patterns and rainfall probabilities must also be considered. Rotation intensity is the demand

for water created by the rotation. Growing crops that use high amounts of water will increase intensity. The level of intensity should complement the water supply. Hence, zero-till rotations are generally more intensive than conventional-till rotations. In dry areas a mix of high and low water use crops should be grown. In more humid areas more high water use crops should be selected. Soils with high water holding capacity (i.e. clay soils) support greater rotation intensity than coarse textured soils. If fields are consistently too moist, the current rotation lacks intensity. If fields are too dry, rotation intensity is too high. If the extra water in a no-till soil is not utilized, there is greater risk of insufficient nutrient utilization and environmental risk..

Any crop that is using soil moisture into September is depleting available soil moisture reserve for next year's crop [Domitruk et al., 1997]. Apply herbicides to desiccate crops such as lentils and flax can prevent them from unnecessarily utilizing valuable moisture. However, this is no reason to eliminate long season crops from a rotation. Long season crops utilize excess nitrogen stored deep in the soil profile thereby preventing leaching of nitrates. Soil moisture and seedbed conditions are better in conservation farming systems because more snow is trapped in standing stubble, and moisture and surface residue losses due to tillage are reduced. These factors join to increase the amount of water entering the soil and to reduce subsequent evaporation losses.

Well planned crop rotations are the key to successful direct seeding systems. Rotations are an effective method of reducing problems with diseases and weeds by breaking pest cycles. Weed control can be maximized by utilizing the weed control strengths of different crops. These strengths include competitiveness, life cycle and herbicide compatibility. The most successful direct seeding systems minimize or eliminate summerfallow. All crop rotations involve compromises. The challenge is to plan crop rotations that maximize the benefits and minimize the risks.

2.1.2.3 Pest Control

Control of pests in crops is usually focused on plant disease, insects, and weeds. Pest management in conservation farming systems is complex and requires more foresight, information and knowledge.

2.1.2.3.1 Disease Control

The presence of any plant disease is dependent on the existence of the plant disease triangle, which consists of a pathogen being present, a suitable environment, and the presence of a susceptible host [PAMI, 1999]. The best disease management programs utilize a method that eliminates one of the required disease conditions and thereby prevents the disease. Crop rotation is an effective method of accomplishing this. In areas and in years with higher rainfall, crop rotation and tillage management will have an effect on yield losses associated with disease. Although zero tillage normally results in higher levels of foliar disease, the effect upon root disease depends on the disease crop combination involved. Research indicates that in most years improved plant growth under zero tillage enables plants to compensate for higher levels of disease, resulting in losses that are generally only slightly higher than under conventional tillage [PAMI, 1999]. Crop rotation is an effective method of breaking the disease cycle and thereby should always be included as part of the disease management strategy [Bailey et al., 1995]. However, some diseases such as wheat stem rust are not affected by crop rotation and thus cannot be controlled in this way. Wheat stem rust does not overwinter on residue or in the soil, but spores are carried into Canada by winds from the southern states and is controlled through resistant cereal cultivars.

Most pathogens survive on crop residues over winter. Crop residues left on the soil surface decompose slower than those buried. Tillage buries the crop residue and aerates the soil increasing the decomposition rate of plant residue reducing pathogen levels. Consequently, direct seeding has the potential to increase the levels of some diseases by promoting the survival of pathogens. Furthermore, elevated soil moisture conditions in untilled fields might enhance the environment for the development of some plant diseases [Domitruk et al., 1997]. Crop rotation is crucial in direct seeding systems as it is one of the few disease management methods available to producers. Of equal importance, growing crops for a few years that are not susceptible to disease lowers soil and residue-borne disease levels. Although some diseases can survive on dead plant tissues, they require a living host to multiply. By rotating crops, the reproductive cycle of the disease organisms is broken. This helps reduce the amount of disease inoculum present, thereby reducing

disease when a susceptible crop is grown. As some pathogens can remain viable in the soil for several years, a break of several years may be required before growing a susceptible crop in order to effectively control disease. In general, susceptible crops should be spaced as many years apart as possible. For example, to minimize blackleg, canola crops should be spaced at least four years apart, even if relatively resistant varieties are being used.

Fusarium Head Blight in cereal crops has become epidemic in areas of Manitoba and problematic in eastern Saskatchewan during years with hot moist conditions, particularly on clay soils where direct seeding retains cereal stubble [Tekauz, 1997]. Currently, the best defense is crop rotation with as few cereal crops as possible. Barley is usually less susceptible than wheat, which can experience losses up to 35% [PAMI, 1999].

Presence of disease on adjacent land can cause problems when planning rotations. Infection can be caused by wind-blown residue and spores even if long rotations are being used. This problem is prevalent with blackleg in canola and flax rust. Weeds and especially volunteer crop plants, can act as a “green bridge” for plant diseases. Hence, it is important to control volunteer host plants to achieve effective disease control using crop rotations. Take-all disease in winter wheat, blackleg in canola and rhizoctonia root rot in lentil are all examples of diseases known to cause problems after volunteer crops [PAMI, 1999].

The easiest method to control or reduce seed-borne diseases is to use disease-free seed or disease-resistant crop varieties. Seed treatment fungicides are also an effective method of controlling many diseases.

Recent research shows that chaff can cause a toxic effect on the following crop due to increased levels of root diseases [PAMI, 1999]. Effectively spreading chaff can help lower disease levels by reducing volunteer crop plants and eliminating excess moisture under the chaff mat where moist cool conditions are ideal for root diseases such as take-all and pythium root rot. In addition, volunteer crop plants and other weeds are also usually worse in the chaff row where they serve as hosts for pathogens that will attack the emerging seedlings. Significant yield reductions result from disease in this manner if chaff is not spread effectively.

2.1.2.3.2 Weed Control

Of all the agronomic problems that farmers must solve and make action decisions, weeds are the most common and perhaps the most challenging. The types and densities of weeds found in fields are determined by local soil and climate conditions, and by specific management factors such as tillage, herbicide use and cropping practices. Because of the interrelationships that exist, when one management factor changes, such as converting from conventional to conservation tillage, changes must occur in other factors to ensure effective weed control. With the abundance of available herbicides in recent years, producers have become dependent on herbicides as the primary method of weed control. Normally, the procedure is to identify the weed problem, estimate the damage potential, and decide whether or not to apply a herbicide. This process fails to recognize the interrelationships in a conservation farming system, such as the reasons for changes in weed populations, or to analyze why weeds have become a problem.

An integrated management approach examines the overall management of weed communities; it is not a recipe of concise control measures to implement during outbreaks. Rather, the goal is to devise a system of related practices that will encourage crop development over weed development, creating an environment where weed growth is reduced. Integrated weed management uses a combination of numerous weed control measures including prevention, sanitation, cultural, biological, physical and chemical methods [Domitruk et al., 1997]. Although no one control measure can be expected to provide acceptable levels of weed control on its own, if weed control measures are utilized in a systematic and coordinated manner significant advances in weed control are achieved.

An integrated weed control program has four basic components which apply to any cropping system, but they are particularly important in conservation tillage systems where management changes are inevitable [PAMI, 1999]. These components are described in the following discussion.

Familiarity with the kinds and numbers of weeds in a field and their growth stage, relative to the crop development stage, is important. In addition, accurate records are essential to enable future planning of crop rotations and weed control strategies. These records should include previous crops (i.e. rotations), herbicides used, soil fertility, and

weed histories so that new weed species are easily identified before they invade large areas of the fields.

Understanding the biology of weeds is essential in implementing an integrated weed control program. Knowledge required includes weed types (i.e. life cycle) such as summer annuals, winter annuals, biennials, perennials and their method of reproduction, requirements for germination and emergence, time of flowering, duration of seed dormancy, and control strategies for each weed.

The weed population in a field is never constant and adapts to certain environmental conditions, crop types, and agricultural practices. For example, annual weeds such as wild oat and green foxtail have adapted to yearly spring cropping practices, making their control the major herbicide expenditure in annual crop production. Another example of weed-cropping system interactions is when some weed populations become tolerant and can withstand particular herbicides or herbicide groups. This herbicide resistance is a result of continuous use of specific herbicides that kill weeds in the same way. This problem has been enhanced by poor cropping practices including the continuous planting of the same crop on the same field. In conservation farming systems, weed-cropping interactions are of particular importance because when tillage is reduced, producers must optimize the effectiveness of their cropping systems in order to achieve satisfactory weed control without increased reliance on herbicide use.

An integrated weed management program selects the most appropriate combination of control measures that will provide effective and economical control without harming the environment. Across the prairies, different regions experience a variety of weed populations, climate, soil types, and cropping practices, and with conditions varying from farm to farm it is essential that producers develop their own integrated weed control programs [Domitruk et al., 1997]. Some of the control measures include:

- 1) Preventing the introduction of weeds into an area by using good quality seed which maximizes the competitiveness of the crop, by cleaning the seed to remove weed seeds and smaller seeds which produce less vigorous and competitive seedlings, and by taking care that the movement of farm machinery and the transportation of crops to market do not increase the risk of

spreading weeds. Many weed seeds can also remain viable after passing through the digestive system of farm animals and thus spreading manure on crop land is also a potential source of new weed infestations.

- 2) Sanitation which refers to preventing the spread of existing weeds to new areas and includes controlling weed communities in non-crop areas such as fence lines, road allowances, stone piles, shelter-belts, and hay and pasture areas adjacent to cultivated fields.
- 3) Cultural controls which include any practice that increases the competitive advantage of the crop over weeds. Dense vigorous crop stands will reduce the risk of lower yields due to weeds as will ensuring that the crop has a good head start on any weeds that germinate. Examples of cultural controls include selecting crops that grow rapidly, producing tall plants with large leaf areas to compete aggressively with weeds. As previously mentioned, developing the most appropriate crop rotation is important in implementing an effective weed control program, but it is also very challenging because of the interrelations with other factors such as available moisture, diseases, insects, fertility, markets, crop residues and environmental parameters.

When developing crop rotations, the history of weeds in fields is important so that it is known what weeds are present, how they affect the control options for the crop to be grown and how effectively the crop will compete with the weeds. Knowing the history of herbicide use is also advantageous because if there are herbicide residues present, the choice of crops is limited. Also beneficial, is knowing whether the crop to be seeded allows for in-crop control of the weeds present and if the proposed crop rotation allows for the use of different groups of herbicides to reduce the potential risk of developing herbicide resistant weeds.

The field's soil fertility history is important to ensure that adequate nutrients are present to promote vigorous crop growth and to determine whether the fertilization practices such as rates and placement favor the crop and not the weeds. Knowledge of the history of crops (i.e. rotations) is needed to determine whether volunteer growth from the previous crop needs to be controlled and if so, do the proposed crops allow for adequate

control. Also advantageous to know, is whether insects or diseases affected the previous crop, as these will limit cropping alternatives for the current year, and how have previous crops affected the weed populations. Guidelines for developing crop rotations are [PAMI, 1999]:

- 1) use competitive crops such as barley and forages in rotation to control weeds that are difficult to control through other means;
- 2) rotate between spring and winter annual crops to disrupt weed life cycles;
- 3) alternate between different crop types in rotations such as cereals, oilseeds, pulse crops and forages to allow for a broader spectrum of in-crop weed control options and to prevent a build up of specific weed species;
- 4) avoid planting a crop back into its own residue as this will promote an increase in the occurrence of disease and insect problems. The presence of disease and insects will stress the crop making it less competitive and more prone to weed infestation.

Earlier seeding dates enable crops to capitalize on available soil moisture, sunlight energy and nutrients before weed populations have a chance to establish. However, seeding dates vary depending on the crop and variety to be grown, the regional climatic conditions, and the soil temperature [Domitruk et al., 1997].

Increasing normal seeding rates by approximately 25% increases crop competition against weeds and improves yield [PAMI, 1999]. Although increasing the seeding rate of cereals results in a decrease in the number of heads per plant, the number of spikes and the density of spikes increase as the number of established plants increase resulting in overall higher yields. Row spacing and soil disturbance determine the amount of weed growth that occurs because soil disturbance promotes weed growth particularly between the seed rows. Naturally, narrow row spacing will result in more soil disturbance than wider row spacing but soil disturbance is also interrelated with the type of opener used. That is, a seeder with narrow hoe or disc openers will not disturb the soil between seed rows as much as sweeps or split-boot openers.

As with all cultural practices, seeding depth should be maintained to optimize the competitive advantage that the crop has over weeds. Seeding depth does not refer to the

depth setting of the seeder but rather to the amount of soil cover over the seed. Planting at a uniform shallow depth into adequate soil moisture will encourage rapid germination and growth of the crop preventing weeds from emerging before the crop and minimizing the need for in-crop weed control. Although in a conventional farming system it is common to “seed to moisture”, that is, to increase the planting depth under dry surface soil conditions, this practice delays crop seedling emergence and the crop’s ability to compete with weeds. In a conservation farming system, rapid crop establishment is easily achieved through optimal seeding depth due to a firm seedbed and generally adequate soil moisture.

One goal of the seeding operation is to establish a favorable environment where crop seedlings have a competitive edge over weeds. Packing the seed rows improves seed germination and plant emergence by maximizing seed-to-soil contact, delaying soil drying above and around the seed, and by improving the movement of water to the seed. For the same reasons that packing the seed rows benefits germination, the opposite is true for the weed seeds between the rows due to poor weed seed-to-soil contact. The soil between the seed rows is largely undisturbed by the seeding operation and remains so throughout the growing season. When harrow or harrow-packer bars with coil packers are used, they have the opposite effect, where the entire soil surface is disturbed and packed following the seeding operation creating an ideal seedbed for weed germination.

To compete with weeds, a vigorous crop growth depends on an adequate supply of plant nutrients. Essential to the success of conservation farming is a good soil fertility program built on crop rotations and fertilization. Required nutrients must be supplied to the crops in the proper balance. Regular soil tests indicate the amount of macro-nutrients required for crop growth. In addition to fertilizer applications, effective rotations should include legume crops, if possible, as they contribute significant quantities of nitrogen to the soil and improve soil quality.

Weeds will respond to crop nutrients in the same way as crops. Hence, fertilizer should be applied so that it aids the competitiveness of the crops and not the weeds. Nitrogen is mobile and hence should be seed placed and/or banded close to the seed where it will not be readily available to weed seeds. However, producers need to exercise caution when applying fertilizer in conservation tillage systems because with the reduction, and

ideally, elimination, of tillage, direct seeding is often employed where fertilization and seeding are combined in one pass. Hence, all the required fertilizer is applied at seeding time either with the seed, banded below the seed, or a combination of the two. Excessive nitrogen fertilizer placed with the seed will damage young crop seedlings and reduce plant growth. As a result, there is an opportunity for higher weed infestations which reduce crop yield. Kirkland's research in "New Developments in Direct Seeding Weed Management" (unpublished) found that side banding nitrogen is more effective in reducing weed densities and biomass than is pre-seed banding. O'Donovan et al. (1985) showed that when nitrogen was banded, the green foxtail population decreased as the rate of nitrogen increased, as was also the case with stinkweed. In addition, green foxtail populations were shown to be reduced more under zero tillage than conventional tillage. A study by Blackshaw et al. (1994) found that deep banded nitrogen reduced foxtail barley biomass and increased wheat yields as compared to surface applied nitrogen.

In some cases, prevention, sanitation and cultural controls can not provide effective weed management, especially in areas with patches of weed growth. Sometimes tillage, with an implement that minimizes residue loss, is the only method to effectively control weeds. A tillage operation at slow speed on dry soil in dry weather with a cultivator equipped with low crown sweeps, or wide blades, or with a trailing rod attachment, or a rod weeder, will provide opportunity to eradicate stubborn weed problems. Although mowing is used for controlling weeds in ditches, road allowances and fence lines, it can also be used effectively in conservation tillage, particularly if conservation fallow is part of the rotation. However, proper timing is critical to the success of mowing operations where the type of weeds and their flowering time helps determine the optimal time for mowing weeds before they set seed. Chaff management might also be necessary to control weeds. Any weed seeds or grain not recovered by the combine at harvest time are spread back in the field along with the chaff. Although in drier areas the chaff layer is an advantage for retaining moisture, in moister areas the chaff can present some problems in that some of the weed and crop seeds will germinate causing weed problems in the future. If chaff is not spread evenly over the entire field it can exhibit a toxic effect and retard germination of subsequent crops in the rotation.

Today, chemical weed control plays a large role in a producer's integrated weed management control program. In some instances herbicides are used when the level of weed infestation does not justify treatment. This practice is not cost effective as the increase in crop yield will not compensate the cost of herbicide application. Agronomists recommend that to maximize profits, the use of herbicides be minimized and used only to supplement prevention, sanitation, cultural and physical control measures. Surveys have shown that weeds will always be present in a field regardless of the farming system used. The challenge is to determine when weeds will begin to cause yield losses and by how much, which is the key to determining threshold levels. It is economically important to determine whether the weed population surpasses the threshold level at which it starts to affect factors such as crop yield, ease of harvest, dockage, crop quality and marketability, and subsequent weed populations. Determining threshold values requires knowledge of the following:

- 1) **Weed/crop Competition** - Herbicide use may not be cost effective in crops such as barley and rye that have a competitive advantage over weeds.
- 2) **Weed Density** - as weed densities increase there is a greater likelihood of yield losses although the level of density causing yield reduction will vary depending upon the climatic conditions and the soil zone.
- 3) **Effective Time of Weed Removal** - To reduce crop losses due to competition, weeds must be removed early. Harker et al. [PAMI, 1999] found that yields of broadleaf crops increased the earlier weeds were removed. The longer weeds are permitted to compete with the crop, the greater the potential crop yield loss will be.
- 4) **Effective Time of Emergence** - Because competition is closely related to relative time of emergence of the crop and weeds, crop losses are significantly lower when the crop emerges before the weeds.
- 5) **Crop Rotation** - If the subsequent crop in the rotation is a poor competitor, or if it has limited in-crop weed control options, it may be beneficial to use herbicides to control the weed in the current crop.

- 6) **Commodity Prices/Herbicide Cost** - The cost of the herbicide used in a crop with low market value may not be economical. However, in some instances, merely the presence of weed seeds in a sample can devalue the crop. For example, wild mustard and/or cleaver weed seeds in a canola sample can lower the grade even though the canola may not suffer any reduction in yield.
- 7) **Herbicide Performance** - There are several factors which influence the performance of herbicides including:
 - **Timing:** Weeds are easier to control when they are young and actively growing. In addition, early removal stops competition, saves moisture, and increases crop yields.
 - **Growing Conditions:** Herbicides are more effective when growing conditions are favorable. Weed control is more effective when herbicides are applied during the cooler parts of the day. Otherwise, high temperatures and low relative humidity result in rapid drying of herbicide droplets and increased evaporation losses.
 - **Soil Factors:** The effectiveness of soil active herbicides is best when soil moisture conditions are good as granular formulations require more soil moisture than liquid formulations. Soils high in organic matter and/or clay will require higher rates of herbicide to obtain adequate control.
 - **Other factors** that effect herbicide performance include water cleanliness, water hardness, water volume, and equipment. However, these factors depend on the quality of the spray water used and routine maintenance of equipment, and are outside the realm of a decision support system.

Weed control is the most apparent difference between conventional farming practices and conservation tillage systems as the latter requires more management, better field monitoring, and a good understanding of herbicides. The goal of chemical weed control in conservation tillage is to ensure that the crop has a competitive edge over weeds.

2.1.2.3.3 Insect Control

Insects in conservation farming systems are controlled in a similar manner as disease. Unincorporated crop residues have the potential to increase not only disease but

insect problems as well. The effects these problems can cause are directly related to crop stress. Conservation tillage can improve soil structure and moisture availability and thereby reduce crop stress resulting in positive net result. Crop rotations containing a variety of crops interrupts the life cycle of many insects and provides opportunities for in-crop control using an insecticide.

2.1.2.4 Residue Management

Residue management or trash management is the residue handling practices required to complement the direct seeding and crop production process. Residue management is a critical component of a direct seeding system and begins at harvest where failure to properly handle crop residue often results in seeding problems and poor crops the next year.

As discussed, crop residue is an overall benefit to direct seeding systems, but unmanaged residue can cause many problems. Initial problems include plugging and/or hairpinning with subsequent problems including poor or uneven emergence, cold wet soils, nutrient tie-up, and delayed and uneven crop maturity. As mentioned earlier, concentrated chaff may also have a toxic effect on seedlings which results in thin plant stands in the chaff rows. For example, disc drills may have difficulty seeding through heavy residue, especially chaff residue, whereas hoe drills and shank systems may plug in heavy, poorly managed residues particularly those with concentrations of long straw. Factors which affect residue management are as follows:

- 1) Crop Type - has a direct effect on the type and amount of residue, and the management needed for direct seeding. For example, wheat, barley, durum, rye, oat, and long vine pea crops may produce large quantities of straw and chaff which must be managed before direct seeding is possible. The crop variety chosen can also dictate the amount of residue produced. Henry and Bulman [PAMI, 1999] found that different varieties of wheat based on a 35 bu/acre yield, produce different quantities of both straw and chaff. Alternatively, canola and mustard produce very little straw but large quantities of chaff. It is the chaff from these crops which produces toxins that inhibit the growth and development of seedlings. Hence, good chaff spreading is

essential. Lentil crops normally do not result in residue management problems, but rather may not produce sufficient residue for some purposes such as snow trapping.

- 2) **Width of Cut** - of the combine header or swather has a direct effect on the type of residue management employed with additional management being required as the widths increase. For example, a swather width of 45 feet produces a straw concentration three times as heavy as it would be if compared to a full width spread [PAMI, 1999]. With an 80 bu/ac barley crop the straw concentration in the spread pattern is equal to the straw production in a 240 bu/ac crop. Even the highest clearance seeding equipment will not do an adequate job of seeding through this amount of residue. Similarly, the combine's chaff row will be much heavier when compared to a small header. If straight cutting, the same result occurs, but if cutting height is higher than with swathing, straw management is reduced proportionately. Longer stubble can cause problems as the height increases. Today, straw and chaff spreading equipment can effectively handle the widest widths of cut whether from a swather or header.
- 3) **Stubble Height** - after harvest has several effects on a direct seeding system including snow trapping capability, reduction of evaporation, and the capability of seeding equipment to pass through stubble. In Saskatchewan, snow represents one quarter to one third of annual precipitation with snow trapping increasing yields by 10% or more [de Jong et al., 1986]. Snow can be trapped by tall stubble, alternate height stubble, and trap strips. Tall stubble ranges from 12 to 24 inches and alternate height stubble varies from short 6 to 12 inches, to tall stubble 12 to 24 inches high, in alternate swather passes. Trap strips are narrow strips of tall stubble running through short stubble. Standing stubble can also have a positive effect on the following crop in the rotation as wind velocity is reduced at ground level and evaporation is reduced by about 40% from wet soil surfaces compared to bare soil surfaces [Caprio, 1986]. These conditions can reduce plant stress and

may result in higher yields compared to tilled stubble [Cutforth and McConkey, 1997]. Hoe and shank type direct seeding equipment can experience problems in excessively tall stubble in that, in spite of clearance, tall standing stubble will wrap around the shanks and plug the machine. Compounding this problem is high moisture content in the straw which often occurs in the mornings or evenings. For sweeps or equipment with shank mount packers that reduce residue clearance, the rule of thumb is that the stubble height should not exceed one times the row space of the seeder. For drills with narrow openers, the rule of thumb is that the stubble height should not exceed 1.5 times the row space of the equipment. Alternate swath heights and trap strips can cause plugging problems even if seeding direction is angled to the strips. This problem is not as significant with narrow trap strips as compared to alternate height swathing. Disc type seeding equipment has fewer seeding problems in tall standing straw as there is less residue on the soil surface to cause hairpinning. However, lodged straw will cause problems for both disc and shank type seeding equipment. Although high standing stubble is most beneficial for snow trapping there must be a compromise between stubble height and trash clearance to prevent plugging problems during seeding. To avoid seeding problems, stubble and strip heights should be limited to that which is best cleared by the seeding equipment.

- 4) Swathed versus Straight Cut versus Stripper Header - The method of cutting does not affect residue management, except stubble height may be taller with a straight cut system. Chaff quantities will be the same with either system although straw quantity will be much higher with the lower cutting height of the swather because of the higher straw to grain ratio. A high stubble straight cut system will produce less straw resulting in less straw management required to provide good seeding conditions. Although the use of a stripper header will increase harvest speed, the residue left in the field must be manageable. Hoe drills are not as effective in tall stubble as disc machines

which work well as long as the stubble remains standing (i.e. anchored residue). For instance, if a heavy wet snow lays the stubble on the ground, discs will not be able to handle the residue.

- 5) **Straw Quantity** - The type of crop, variety, and environmental conditions determines the quantity of straw produced by the crop. Cereal grains such as barley, wheat, rye and oat usually produce large quantities of straw which normally do not break down substantially during threshing. Other crops such as canola, mustard, short vine pea, and lentil produce less residue and their straw breaks down during threshing. In general, crops' straw which breaks up during threshing often produces large quantities of chaff, which will require combine attachments to adequately spread the chaff.
- 6) **Conventional versus Rotary Combine** - Rotary combines produce shorter straw which does not require chopping, whereas conventional combines produce long straw and choppers are required to cut it. Additional straw spreading capacity is required for both types of combines with wide swaths or straight cut headers and both systems require chaff spreading in a direct seeding system.
- 7) **Straw Choppers and Spreaders** - Until recently, most original equipment choppers and spreaders performed poorly in spreading straw especially under high yield, wide swath conditions. Now producers can avail of some after-market choppers and spreaders as well as some optional factory equipment which will do an adequate job of spreading straw.
- 8) **Chaff Spreaders** - do not exist on most factory-equipped combines, although some have them available as options. If a chaff spreader is not present, chaff rows and their associated problems may result. Experienced producers using direct seeding systems strongly suggest that good chaff management is critical for successful direct seeding [Evans, 1993]. Hence, the need exists for chaff spreading equipment in direct seeding systems.
- 9) **Harrowing** - Although harrowing does not move chaff, which must be managed at the back of the combine, harrowing can be an effective method of

spreading straw. Harrowing produces the best results immediately after combining although this practice does knock down stubble thus reducing potential snow trap. As harrowing requires labor, fuel and equipment, it adds to the cost of the overall direct seeding system.

2.1.2.5 Seeding and Fertility

A suitable system for seeding and applying fertilizer is essential for the success of a conservation tillage and/or direct seeding system. However, as with the other aspects of conservation tillage discussed thus far, seeding and fertility are complex issues. A successful seeding operation ensures rapid, even seedling germination and development through reduced seedling disease, improved weed competition, early crop maturity, and maximum yield. Factors that affect seedling development include:

- 1) **Seedbed Quality** - is essential to the start of any successful direct seeded crop. In conventional tillage systems, seedbed quality is determined by the amount and type of pre-seed tillage where the depth of cultivation and the amount of soil disturbance will determine the depth and condition of the firm, moist layer of soil that the seed should be placed in by the seeding equipment. Normally, the soil dries out to the depth of tillage so the seed must be placed below the tillage zone in order to reach moisture. The opposite is true in direct seeding systems where seedbed quality is not affected by pre-seed tillage and the firm, moist soil conditions necessary for rapid germination and emergence are located very near the soil surface. Naturally, as the seeding season progresses, some surface soil drying occurs and deeper seed placement is required. Seed must be placed at the proper depth into moist soil to ensure germination occurs and the seedling is established before the soil dries around the seed. An appropriate opener must be selected for soil type and residue management to ensure that the seedbed quality is maintained and the seed and fertilizer are correctly placed.

A healthy soil is a primary factor of a quality seedbed. The fundamental economic viability of any farming system is how it affects the soil over the long term and research has consistently proven that the soil

becomes more healthy, vital and productive with reduced tillage. Reduced tillage increases soil organic matter content, improves soil tilth, increases soil moisture holding capacity and improves soil moisture infiltration rates.

Moisture is the most limiting factor to crop production on the prairies. Although moisture conservation is a primary benefit of direct seeding, it is also a fundamental building block in the development of the direct seeding system. In the process of balancing seeding, fertility and weed control concerns, one can easily forget that protecting seedbed moisture is important. In the past, residue management, weed management, and saving moisture for next year's crop were the reason that prairie farmers adopted high tillage management systems. Today, the successful implementation of a direct seeding system requires that residue management be the priority, and for this reason, agronomists recommend that the first piece of equipment purchased by a direct seeder be a good straw and chaff spreading system for the combine. Heavy concentrations of chaff and straw in field rows can interfere with seedbed quality and in some severe cases, tillage may be necessary before seeding can begin.

- 2) Proper seeding depth is the key to a successful seeding operation and is determined by moisture, soil temperature, and crop type. The resulting seeding depth has a direct effect on the time to emergence, which increases as seeding depth increases. Prairie farmers have often expressed the concern of seeding into cooler soils with direct seeding then with conventional tillage systems. High disturbance tillage buries residue resulting in a black soil surface which absorbs more sunlight than residue covered zero-tilled fields. Spring soil temperatures of direct seeded soils can be about 1°C or 2°C lower than tilled soils which may delay emergence by one or two days, but not crop maturity [PAMI, 1999]. A rule-of-thumb is that the time to emergence of seedlings halves for every 5°C increase in soil temperatures between 5°C and 20°C [Lafond and Baker, 1986]. Lafond [PAMI, 1999] indicates that soil

temperature with direct seeding is not a serious concern if good agronomic practices are followed.

- 3) **Packing** is the compaction of soil after seed placement to increase soil bulk density and to invoke changes to the soil that benefit germination and emergence. The germination of seed requires water absorption and greater absorption occurs when water contacts the entire seed. Packing increases soil density which indirectly enhances seed-water interchange mechanisms through a number of interrelationships. The compaction of soil around the seed reduces soil porosity and soil aggregate size resulting in improved seed-soil contact. A decrease in soil porosity reduces capillary size which enhances the movement of water to the seed from deeper, high moisture areas. Hence, if moisture is available below the seed, packing will improve seedling emergence. Excessive packing in heavy, wet soils forms a crust layer on the surface and can affect emergence. Hence, packing at the surface above the seed can retard emergence especially in wet, heavy soils and Gray Wooded soils which are low in organic matter, whereas packing at the seed level improves germination.

The two main methods of packing are on-row packing and random packing. On-row packing is achieved by press wheels mounted at the rear of the seeder or by shank-mounted packers. Random packing is accomplished by coil packers, crowfoot packers, or other methods that randomly pack the surface area without respect to seed row location. On-row packing best achieves the needs of direct seeding by packing the seeded area without disturbing adjacent soil. Hence, this packing method does not promote weed growth, pulverize the non-seeded soil surface, or provide uneven seed row depth as a result of uneven packing. Instead, the groove formed by the packer wheel reduces seed depth, providing a more uniform seed depth and a microenvironment, which protects the emerging seedling. Harrow packer drawbars do not provide the same local microenvironment and do pack adjacent areas causing weed seed germination between the rows. Although

packing improves germination and emergence, heavy packing weights increase draft and decrease energy efficiency. Large diameter wheels and a wheel width relative to the seed row width minimize draft and increase efficiency. The interrelated soil mechanisms of packing in a direct seeding system are not well understood. Though a joint project, the Prairie Agricultural Machinery Institute and AAFC are conducting a detailed study examining packer weight and packer wheel type with three types of seeding tools commonly used in direct seeding systems.

2.1.2.5.1 Fertility

Correcting soil nutrient deficiencies is essential in order to achieve maximum economic yields. Nitrogen usually becomes depleted in all continuous cropping systems including direct seeding systems. There are many types of nitrogen fertilizer with anhydrous ammonia, liquid and granular nitrogen (ammonium nitrate and urea) being the common nitrogen sources. The many methods of application contribute to the complexity of conservation farming systems and include fall or spring banding, side banding, seed placing, mid-row banding, nesting, below seed row banding, fall or spring broadcasting and post-emergent broadcasting, dribbling, or spraying. The nitrogen source selected must conform to the method of application and timing, local availability, cost, farmer preferences, and type of seeding equipment. Important alternatives to consider are swine manure and legumes in rotation which can supplement or provide the necessary fertility requirements.

Placing the fertilizer with the seed is also used but is limited to the quantity of nitrogen that can be placed before germination. Alternatively, side banding, mid-row, and below-row banding are all “one pass” systems which usually result in efficient nitrogen use. A major concern of these application methods is that all required fertilizer is placed during the seeding operation. Sometimes it is difficult to place all the required nitrogen, particularly if high rates are required. Alternatively, banding systems are advantageous as they reduce field operations, provided that there is adequate separation between the seed and the fertilizer. The width of separation within the seed rows has a major effect on the amount of seed placed fertilizer that can be tolerated by a given crop before negative effects

occur. Also, in good soil moisture conditions higher rates of fertilizer can be placed with the seed compared to dry soil conditions. Mid-row banding, where fertilizer is placed between every second seed row, has the advantage of not disturbing the seedbed by fertilizer placement. One additional concern with these three methods of banding is the high power requirements. With the fertilizer being placed below the seeding depth, more tractor horsepower is required to place the fertilizer and seed at the same time than would be required if seeding only. It is important that the crop be fertilized and not the soil in order to achieve fertilizer use efficiency.

There is a large amount of information related to direct seeding equipment and openers, however, since this chapter is focused on describing the complexity of conservation farming systems, such level of detail is not required for the reader to gain this appreciation.

2.1.2.6 Equipment and Economics

Two basic areas of concern when contemplating a direct seeding system are equipment requirements and the economics of a chosen system. Each area has a large number of interrelated options linked to crop choices, rotations and soil types. The best strategy is to start with equipment decisions and then analyze the economics for each scenario modifying the plan or scenario until the equipment costs fit with the current economic realities of agriculture.

2.1.2.6.1 Equipment

Agricultural dealers have excellent direct seeding equipment available as well as combines with high quality straw and chaff spreading capabilities. If the immediate scale-up of equipment investment cannot be economically justified, there are many inexpensive equipment alternatives to enable a farmer to begin direct seeding. Starting with a lower capital equipment investment has a major advantage when converting to direct seeding in that if the initial equipment selection does not fulfill the specific conditions of the farming system, a change in machinery is not a major obstacle. Throughout the preceding discussion in this chapter, there have been many references to the high level of complexity involved in conservation tillage and direct seeding systems. The selection of equipment is

no less complex. The following are examples of questions that must be answered as part of the equipment decision analysis.

- 1) If fallow is being used as a means of handling residue, controlling disease, and pests, and tillage is being used in fall and spring for weed control and incorporation of herbicides and fertilizer, then does the rotation need to be extended to provide better opportunities for weed control? What new crops will be added to the rotation and what machinery will be required? Will continuous cropping be implemented to avail of the higher moisture reserves in spring?
- 2) Is straw chopped and spread, along with chaff, adequately to allow direct seeding? What stubble height is desired? All residue should be spread at the combine to avoid seeding problems or having to harrow poorly spread straw. Chaff cannot be spread by harrowing; all residue needs to be spread at one time.
- 3) Direct seeding systems require more herbicide applications and a more flexible application system. Pre-harvest in-crop herbicide applications are becoming important for controlling persistent perennials. Is a high clearance sprayer or custom application the right decision for some or all of the spraying and if a sprayer is purchased will it operate in the rough field conditions created with direct seeding?
- 4) What options are available for fertilization; liquid fertilizer? Is anhydrous being fall or spring banded as a separate operation and is this acceptable in a new direct seeding system? Although it is one more field operation, it results in easier seeding and simpler seeding equipment or is side banding anhydrous at seeding time required? Or could all fertilizer requirements be placed by utilizing a wider seed row?
- 5) With the low soil disturbance associated with direct seeding's single pass, it can be an operation which results in rough fields, making spraying and harvesting uncomfortable. When choosing equipment, field roughness must

be manageable in the long run. Soil type, ground speed, type of opener, and average local amounts of rainfall are factors which influence field roughness.

- 6) The row spacing of the equipment needs to be considered. If the crop is swathed, a narrower row spacing may be required for support of the swath. However, wider row spacing equipment is less costly and provides better residue clearance, but may lower the crop's ability to compete with weeds.
- 7) It is very important to consider the horsepower required to pull the direct seeding equipment. However, since only one pass is being made in a direct seeding operation, there is generally more time, and hence a narrow width machine with a small tractor might be chosen or perhaps wider equipment is desired in order to add more acres to the farm operation.
- 8) Sometimes a modification to an existing seeder will allow direct seeding. If not, the producer will need to sell existing equipment and purchase new equipment. This is an easier decision if the new seeder will also work with other levels of tillage, if they should be necessary.

2.1.2.6.2 Economics

In the last decade, about 30% of the annual cropland on the Prairies has been converted to some form of reduced or zero tillage system [PAMI, 1999]. Producers who have adopted these soil tillage practices have reported favorable agronomic benefits. Research has been conducted comparing the economic performance of conventional, minimum, and zero tillage or direct seeding systems. Compared to conventional tillage, conservation tillage provides savings in costs for labor, fuel, machine repair and machinery overhead, but has greater expenditures for pesticides. Machine related savings are a result of fewer trips across the field, combining multiple operations into one, such as seeding and fertilizing, and using equipment with greater capacity and lower draft requirements such as a sprayer rather than a cultivator. Machine overhead savings result from eliminating the need for tillage machines such as cultivators, using smaller tractors with lower capital investment and extending the life of equipment because of reduced annual use. The savings in labor costs is a result of less time spent performing field operations which translates into less hired labor, more free time to possibly expand the farm's landbase

taking advantage of economics of scale, undertaking new or additional on-farm value-added activities, performing custom work for neighbors, working off the farm, and/or more time for farm management and family related activities [Brown et al., 1994]. The demand for time spent on farm management aspects increases with conservation tillage systems because of its complexity and the increased number of decisions to make.

Zentner et al. (1996b) states that in the dry Brown soil zone at Swift Current Saskatchewan, the use of conservation tillage practices in fallow-wheat rotations resulted in machinery and labour cost savings that averaged \$1.20 to \$2.00/ac with minimum tillage and \$2.40 to \$3.60/ac with zero tillage practices, but for continuous wheat systems these cost savings were only about \$0.80/ac. In contrast, Zentner et al. (1996b) reported that costs for herbicides increased by \$4.40/ac with minimum tillage and \$12.40/ac with zero tillage in fallow-wheat areas. The net outcome was that total production costs for the zero tillage fallow-wheat systems averaged 29% higher (\$14.97/ac more) than for comparable conventional tillage managed areas on a silt loam and 14% higher (\$6.80/ac more) on a clay soil. Similarly, total production costs for zero tillage with continuous wheat averaged 10% to 13% higher (about \$9.71/ac more) than for conventional tillage with continuous wheat.

In the moist Dark Brown soil zone at Scott, Saskatchewan, Zentner et al. (1992) found similar results. The machinery and labour cost savings were estimated at \$4.80 to \$5.56/ac for zero tillage with fallow-oilseed-wheat and oilseed-wheat-wheat rotations, but again, the savings were negated by higher herbicide costs. The net result was that total costs for the zero tillage systems, compared to conventional tillage averaged 23% higher (\$13.60/ac more) in the fallow-oilseed-wheat rotation and 8% higher (\$6.46/ac more) in oilseed-wheat-wheat rotation. Smith et al. (1996) also reported that at Lethbridge, Alberta, total costs were highest for zero tillage, less for minimum tillage, and lowest for conventional tillage for wheat and barley production in the Dark Brown soil zone. The high production costs for conservation tillage systems in the Brown and Dark Brown soil zones are a result of the higher cost of controlling weeds on summerfallow with herbicides compared to use of mechanical tillage. There is potential for reducing production costs of conservation tillage practices in these areas by reducing the amount of summerfallow in the rotation, but high evaporation demands, coupled with low and highly variable growing

season precipitation, which are characteristic of these soil zones, makes extended rotations more risky [Zentner and Campbell, 1988; Zentner et al., 1992; Zentner et al., 1996a]. However, snow trapping and using tall stubble or permanent grass barriers can enhance soil moisture conservation and act as a partial substitute for fallow [McConkey et al. 1990, Campbell et al. 1992].

The production costs of conservation tillage systems in the Black and Gray soil zones are much more favorable. Lafond et al. (1993) indicates that in the thin Black soil zone at Indian Head, Saskatchewan, costs for purchased inputs and machinery for minimum tillage and zero tillage were similar, or marginally lower, than for conventional tillage of field pea, flax, spring wheat, and winter wheat production. Although herbicide costs for the spring seeded crops tended to be higher for minimum tillage and zero tillage as compared to conventional tillage, field operation costs had the reverse effect. With winter wheat production, herbicide and machine operation costs were generally the same for all three tillage methods.

The increase in cost savings for machinery and labour from the Brown to the Dark Brown to the Black and Gray soil zones shows the differences in the intensity of tillage used in these areas. For example, farmers in the dry Brown soil zone using conventional tillage practices will till their summerfallow areas for weed control three or four times during the 21-month fallow period. Whereas, farmers in more moist regions using conventional tillage practices might perform six to nine tillage operations on fallow areas for weed control and incorporation of herbicides. Consequently, the potential for savings from substituting herbicides for some or all of these tillage operations is greater in the moister soil-climatic regions than in the drier.

Differences in production costs among methods of tillage practices are only one part of the economic picture. Higher production costs can be accepted if they are balanced by higher revenues from increased grain yields or improved grain quality. In this aspect there is also variation by soil-climatic region where the impact of conservation tillage systems on net returns or profitability also vary with soil texture, depth of topsoil, weather conditions, crops produced, input costs and product prices, and risk preferences [Fox et al., 1991].

In conclusion, it is clear that minimum tillage and zero tillage systems provide improved soil conservation over that obtained with conventional tillage, however, research results on the economics of these systems are less clear. In the dry Brown soil zone, conservation tillage practices were generally less profitable than conventional tillage for both fallow and continuous monoculture cereal cropping systems, even with herbicide costs low. The poor economic performance of conservation tillage practices is a result of a combination of higher production costs (i.e. increased herbicide costs) with no significant yield benefits. It is not known how the economics of conservation tillage systems compare to conventional tillage in mixed or diversified cropping systems in this region.

In the Dark Brown soil zone, zero tillage was marginally more profitable than conventional tillage with mixed oilseed-cereal rotations but only when grain prices were high, or when herbicide costs were reduced modestly from their present levels. However, when grain prices were low, systems which included fallow and used conventional tillage practices resulted in the highest net return.

In contrast, in the Black and Gray soil zones, zero tillage and minimum tillage practices provided higher net returns over conventional tillage for most cropping systems because of significant yield increases and because the costs of production are generally similar for all methods of tillage management in these regions. These research findings emphasize the belief that the greatest economic potential for conservation tillage systems is with extended and diversified crop rotations where the high requirements for herbicides on fallow are avoided.

Direct seeding involves a gradual shift to a new farming system. Producers using fallow as part of their rotation may wish to make this shift by initially seeding into conventional fallow and gradually increasing the intensity and diversity of their cropping system. Cropping intensity relates to the proportion of time that land has a crop growing on it (i.e. moisture use). Less intense rotations are lower risk or more "safe" in terms of producing a crop in dry years. However, more intense rotations have a higher long-term profit potential, even though they may have a higher risk in each individual year.

This chapter has described and illustrated the complexity of conservation farming/tillage systems. The requirement for information and knowledge to solve and

make decisions with regards to this complexity is often satisfied through collection from many sources including Internet sites, government publications, farm management specialists and agronomists. However, in order to effectively utilize this information and knowledge, it must be integrated into an overall farm management system. Often the expertise to assist in this process is limited, due to the time constraints on farm management specialists. Decision support systems assist farm managers in their farm management process.

Chapter 3: Technology Transfer

This chapter reviews the literature and discusses technology transfer and adoption in terms of the adoption process and its characteristics. It also discusses technology adoption with respect to economics, characteristics that influence technology adoption, and concludes with the behavioral influences of technology adoption.

3.1 Sources of Decision Support

The four main sources for decision support (i.e. information and knowledge) include print media (i.e. farm magazines, newspapers, agricultural technical publications and reports), broadcast media (i.e. radio and television), the Internet, and personal contacts. Country Guide, The Western Producer, Farm Light & Power, Report on Farming, Grain News, Top Crop Manager, Canola Guide, Farming, Seed & Biotech, Agro Manager, Seeding, Farm Forum, and Pulse Crop Monitor are publications that are circulated throughout the prairies; most producers subscribe to at least one. Publications such as these, and other printed agricultural information, are generally distributed by post and consequently are not timely. Most radio and television sites broadcast agricultural information at least twice daily and some provide weekly agricultural programs.

In addition to the printed and broadcast media, people such as farm neighbors, elevators agents, seed growers, provincial agricultural extension specialists, federal information service specialists, agricultural product distributors, and farm consultants all convey information to the farm manager in one form or another. For instance, if a farmer tries a new herbicide or fertilization method, his neighbors will naturally ask him how well it worked and other related questions with respect to application and availability. In addition, producer marketing/production clubs are popular where individual farm neighbors meet regularly and share information, knowledge and experiences on crop production.

Grain handling companies such as the Wheat Pool, Pioneer, and Cargill are also farm service and supply centres and sell a variety of farm supplies including pesticides and fertilizer at their grain handling facilities scattered across the prairies. Hence, it is common

for farmers to obtain product information from their farm service agent either via discussion with him or in the form of printed product information.

New grain varieties are licensed by the federal government which provides seed to registered seed growers. The seed growers build up government inspected seed stocks of the new variety. Farmers often buy new grain varieties directly from registered seed growers. Based on their experience, the seed growers will provide information to the producer including recommended growing conditions and how best to sow the seed.

One of the provincial agricultural extension specialist's main duties is to convey information to as many farmers as possible, as well as to be available for consultation about farmers' specific problems. Not surprisingly, the low number of agricultural extension specialists available have a difficult job in servicing all the producers seeking support for their decision making.

Information specialists are located at AAFC research centres. Their goal is to transfer technology as a result of research to the public and provide consultation regarding specific questions or problems through public relations. Both provincial and federal information specialists use the print and broadcast media as well as exhibits and field days to reach the public. Their use of the Internet for disseminating information and knowledge is increasing.

The presence of computer and Internet technology on prairie farms is increasing rapidly. For instance, in Saskatchewan, the presence of computers on farms has increased from 37% in 1992 to 68% in 1998, an increase of 84% [Garven & Associates, 1998]. This trend is expected to continue with approximately 80% of Saskatchewan farms having computers within the next few years [Garven & Associates, 1998]. The number of Internet connections on Saskatchewan farms rose from 16% in the spring of 1997 to 39% in the fall of 1998 [Garven & Associates, 1998]. Surveys indicate that Internet connections on Saskatchewan farms will increase to approximately 60% within the next few years [Garven & Associates, 1998].

More importantly, Garven & Associates (1998) survey indicates for the first time that 17% of Saskatchewan farmers identified the Internet as a potential source of farm management information and 34% of their survey respondents actually used the Internet as

a source of information, up from their 1997 technology survey where 20% of the respondents had used the Internet to access information.

3.2 Technology Adoption

Technology adoption is a process in which individuals accumulate information and knowledge about a technology and then decide whether to use it. With agricultural decision support systems being viable real-world computer applications to transfer technology, and in so doing, help producers make decisions, their success will depend on their rate of adoption, as with any other technology. Targeting the innovators and early adopters as initial end-users, including farmers, extension consultants, agricultural dealers and service agencies, is a good strategy for introducing an agricultural decision support system. The "trickle down" effect is expected to accomplish technology transfer to all potential end-users. Consequently, technology adoption described and discussed in this section is an important factor contributing to the success of agricultural decision support systems and technology transfer.

3.2.1 Categories of Technology Adopters

According to Korsching (1983) there are five categories of adopters:

- 1) Innovators, who form about 2.5% of the population, are high risk takers with less debt and willing to accept more debt. They do not necessarily make the best use of their resources and sometimes fail at their endeavors and/or inventions. They are "venturesome" and eager to try new ideas, communicating with other innovators who may be spread over great geographical distances. They can understand and apply complex technical knowledge;
- 2) Early adopters, who comprise about 13.5% of the population, are decision makers, readily seeking the innovator's technology and when found, fully exploit it. Early adopters are respected, have a high level of credibility, have high social status, and possess a great deal of opinion leadership. They serve as role models and are often the people others converse with before trying a new idea;

- 3) Early majority adopters, who make up about 34% of the population, are also decision makers, but need more time (1 or 2 years) to be shown and convinced of the technology before adopting it. They interact frequently with peers and provide an important link between the early adopters and the late majority in the technology transfer process;
- 4) Late majority adopters, who also constitute about 34% of the population, are risk adverse and skeptical. They must feel totally comfortable with the technology before adopting it and many times it is because of economic necessity and increasing social pressures;
- 5) Laggards, who form about 16% of the population, are last to adopt or may never adopt. They are characterized as "traditional", making decisions in relation to what was done in the past. Laggards are often suspicious of innovations, innovators, and change. Consequently, they are usually the most difficult group for extension specialists to approach, but often the group in greatest need of assistance.

3.2.2 The Adoption Process

Several models of the adoption process have been researched in the past [Lamble, 1984; Beal and Bohlen, 1962; Rogers, 1983; Rogers and Shoemaker, 1968], with the logic being similar in each process. Overall there are basically four steps to the adoption process: acquisition of knowledge of innovation, attributes and persuasion toward an innovation, adoption decisions, and confirmation of decision made. A "trickle down" approach to technology adoption is outlined in Section 3.2.2.5.

3.2.2.1 Acquisition of Knowledge of Innovation

The adoption process begins with the knowledge function, where the potential adopters are first exposed to the existence of an innovative technology and gain some understanding of it. This function continues throughout the decision process as potential adopters acquire more information and understanding of the innovation. In this step, usually three types of knowledge are sought after [Lamble, 1984]:

- 1) Awareness-knowledge consisting of information and consciousness of the existence of an innovation and its main features;

- 2) How-to-knowledge consisting of information and understanding necessary to use or apply an innovation properly;
- 3) Principles-knowledge consisting of understanding the principles underlying the innovation and its use.

3.2.2.2 Attitudes and Persuasion Toward an Innovation

Merely knowing about an innovation does not imply that it will be adopted [Thomsen, 1985]. The potential adopter may not regard the innovation as being relevant or useful. Hence, attitudes about an innovation can modulate linkage between the knowledge and decision functions. Whereas the knowledge function is focused at mainly cognitive mental activity (i.e. knowing), the persuasion function is mainly affective (i.e. feeling), that is, the forming of a favorable or unfavorable attitude toward the innovation.

The persuasion function may be influenced by several factors [Lamble, 1984] including:

- 1) The individual's personality, which might affect his or her perception of the innovation's relative advantage, compatibility and/or complexity within their farming operation, or might affect risk-taking behavior;
- 2) The individual's ability to think hypothetically (i.e. abstractly) in order to apply new innovation ideas to present and anticipated future situations;
- 3) Friends and neighbors might reinforce attitudes toward the innovation, making the potential adopter feel more comfortable about the associated risk, or they might have the opposite effect.

3.2.2.3 Adoption Decision

During the decision phase whether to adopt, the individual evaluates the innovation through activities that lead to a choice to adopt or reject the innovation. One frequent choice is to try the innovation on a trial or probationary basis to determine its suitability in one's actual situation. Depending on the level of advantage or satisfaction gained during the trial, individuals may decide to adopt the innovation/technology.

3.2.2.4 Confirmation of Decision Made

Individuals seek reinforcement throughout the confirmation function for the innovative-decision they have made and attempt to avoid contention. If conflict occurs, it

usually serves as a means to motivate a change in the adopter's attitudes, knowledge or behavior. Furthermore, contention may occur at any stage of the innovation-decision process. For instance, at the knowledge function, an individual may become aware of a felt need or problem. This motivates the person to seek information and knowledge about an innovation which might reduce or eliminate the contention. Similarly, after an innovation has been adopted, an individual may receive further persuasive information which indicates that the better decision was not to adopt. Discontinuing the innovation may reduce the conflict, but on the other side, if reversing the original decision is difficult or impossible, the individual may rationalize his decision by using selective exposure and selective perception to seek only information and knowledge which supports or confirms the original decision. An extension specialist can be helpful in this type of situation by assisting to confirm the individual's desired decision.

3.2.2.5 "Trickle Down" Adoption Process

The "trickle down" process in technology adoption is where technology moves from a small number of innovators through to subsequent adopters. The individuals involved are often subdivided into the five categories of adoption groups described in Section 3.2.1. Four circumstances increase the rate and level of adoption in the "trickle down" process [Thomsen, 1985]:

- 1) introduction of the technology earlier;
- 2) increased size of the innovating group;
- 3) improved quality and speed of information movement between adopters;
- 4) targeted non-adopters.

As more users are exposed to the technology through the "trickle down" process, more individuals adopt or try the innovation and thereby subject it to a greater variety of conditions. Consequently, additional technical modifications and enhancements may be identified and thus the evaluation, trial and adoption stages are subject to further research and development. Of course, not everyone adopts technology — sometimes "tried and true" prevails with individuals reluctant to see any advantages of a new technology that may exist, or society's natural "resistance to change" may contribute to failure to adopt.

3.2.3 Characteristics of Technology Adoption

There are many characteristics governing technology adoption. These characteristics can vary in their relative importance and impact. Four of the major characteristics are discussed in the following sections.

3.2.3.1 Information and Knowledge Sources

A 1983 Alberta Agriculture study concluded that farmers rated printed information (e.g. magazines, newsletters, and government bulletins) as their most useful source of agricultural information because of the ease with which it can be stored, retrieved and referenced [Blackburn, 1986]. Their next most important source of information was neighbors, followed by extension specialists and experts. Furtan's (1981) study in Saskatchewan concluded that the agricultural representative is the most important information source for technical production matters. The greater the perceived risk, the greater the tendency to consult multiple knowledge sources including credible people such as professional agrologists for theoretical and technical advice, and other farmers and family for practical application and personal experiences. Whale (1985) made a distinction between information related to new farming practices and information used for updating. When seeking new ideas and support for decision making, farmers preferred contact with authoritative people [Whale, 1985]. But journals, newspapers and other forms of mass media were preferred for updating information [Whale, 1985].

3.2.3.2 Personal Characteristics

Personal characteristics play an important role in the adoption of technology [Swanson et al., 1986]. Farmers with higher levels of agricultural education tend to more readily analyze technology for its benefits or detriments and to adopt recommended farming practices. With regards to age characteristics, those producers 48 years and older tend to be the adopters while producers age 37 to 44 tend to be the innovators [UTI, 1997b]. Readily available information and concern for environmental issues tend to increase the rate of adoption of technologies related to conservation tillage [Abd-Ella et al., 1981]. Years of farming (increasing the experiential base) speeds decision-making in technology adoption. Hence, it is expected that as more farming experience is gained, technology adoption increases [Abd-Ella et al., 1981]. Family aspirations can be a motive for adopting

technology especially if the technology has financial benefits. It is important that farm family members interact with their environment to learn about available technology, in preparation for possible technology adoption. Technology adoption usually increases with the level of social participation (e.g. "coffee row") and cooperative activity. Off-farm employment provides an additional source of income and social security increasing the farmer's ability to take risk, but reduces the amount of time available for "farming". Consequently, farmers with off-farm employment are more comfortable with higher levels of risk related to technology adoption. Their level of comfort is greater the more the off-farm employment is connected with agriculture [Herrmann and Uttitz, 1990]. Alternatively, off-farm employment may limit technology adoption due to insufficient time to "change" [Boehm, 1995].

3.2.3.3 Farm Characteristics

Farm size is usually the most obvious indicator of the farmer's available economic resources. A larger farm corresponds to a greater ability to take the risk involved in the adoption of recommended practices and technology. Studies have confirmed the positive relationship between farm size and technology adoption [Brown et al., 1976].

Farm diversity may also influence technology adoption as families with less diversified farm operations (i.e. specialized farming operations) normally have fewer technology options to evaluate and consider and therefore there is a greater probability that applicable technology will be adopted readily. Alternatively, diversified farmers, as a result of adopting diversified technologies, will be more able and confident to tackle even more diversification. Demographics can influence technology adoption especially as a result of the "trickle down" process, where the majority of farmers in close proximity to successful early adopters readily adopt technology within localized areas.

3.2.3.4 Farm Structure Factors

The importance of farm structure factors in the adoption of technology cannot be overlooked. Agriculture is now a complex highly competitive industry, dictating the behavior of farmers, where they sometimes are forced to maximize short-term profits at the expense of land resource protection [Swanson et al., 1986]. In order to survive, farmers must increase their scale of farming operation while remaining efficient. Farmers who have

access to land and capital continue to expand their farming operations, whereas those without access are forced out of farming.

The scale of agriculture increased during the 1960s and inflationary 1970s due to market pressures, technological innovation, and relatively low real interest rates [Swanson et al., 1986]. Farmers were able to increase land holdings during the 1970s because of high inflation causing real interest rates to decrease [Sampson, 1985]. However, in the 1980s this situation changed dramatically when real interest rates rose sharply causing farmers with high debt loads, accumulated during the growth era, to suddenly encounter severe financial difficulties. Oversupply of farm products caused low product prices which decreased farm income. This initiated a decline in farm incomes and agricultural land value, increasing the number of farmers with severe economic problems.

3.2.4 Economics

The agricultural economic situation, where the farmer's bottom-line is economic survival, is not always conducive to adopting technology. Farmers are pressured by the competitive nature of the agricultural industry to adopt technology that has short-term payoffs, especially during periods of uncertainty [Margolis, 1977; Rasmussen, 1982]. For instance, with conservation farming technology, farmers cannot be concerned about its adoption and the future of their land resource, if they are not making sufficient profit to cover their immediate costs. Farmers, as individuals, will do whatever is necessary to survive in the "short term", because if their farm does not survive, of course, there is no "long term" for them.

3.2.5 Farm Support Programs

The Canadian federal government has employed capital grants to facilitate farmers' adoption of capital intensive technology such as specialized architectural farm buildings, equipment and machinery. Although tax measures have not been used to encourage investment in technology, capital cost allowance supported the rapid mechanization of agriculture which occurred during the inflationary 1970s [Thomsen, 1985]. More recent support, by the federal government, has focused on the development of demonstration farms, as part of agricultural development transfers to the provinces, to introduce new crops, cropping systems and animal production technology. The Canadian

crop insurance, although not intending to do so, has assisted the transfer of crop technology by guaranteeing a minimum yield and reducing the possible loss from poor performance of an untried technology.

Extension specialists contribute significantly to technology adoption [Lamble, 1984]. One of the major responsibilities of an extension specialist is to facilitate the adoption of new ideas and practices by farmers. In order to be effective in this capacity and accomplish this role, they need an understanding of the processes and factors involved in the transfer and adoption of innovations, of how new ideas and practices are communicated among farmers, and how they decide to adopt or reject the available technology. With this information, the rate of adoption can be explained and predicted, thereby forming a basis for developing effective strategies and planning successful extension programs.

The success of extension programs is often determined by the time lag between the introduction of innovations and their wide spread adoption [Lamble, 1984]. This is sometimes referred to as the "rate of adoption"; the relative speed with which an innovation is introduced and adopted by farmers in the locale.

3.2.6 Characteristics that Influence Technology Adoption

Decision making and technology adoption are influenced by several characteristics discussed in the preceding sections. Figure 3.1 illustrates the decision process for technology adoption linking the various characteristics to show their interaction. Furthermore, the positive and negative influences (i.e. + and - in Figure 3.1) encouraging or discouraging the impact of the different characteristics are also indicated.

3.2.7 Management of Adopted Technologies

Management of adopted technologies will be either from within or through outside services. Which option farmers choose will depend on their time and financial resources, their level of desire for "hands on" management and level of data and information security they wish to attain. Both options are viable and depend on what fits best with the farmer's own agricultural system.

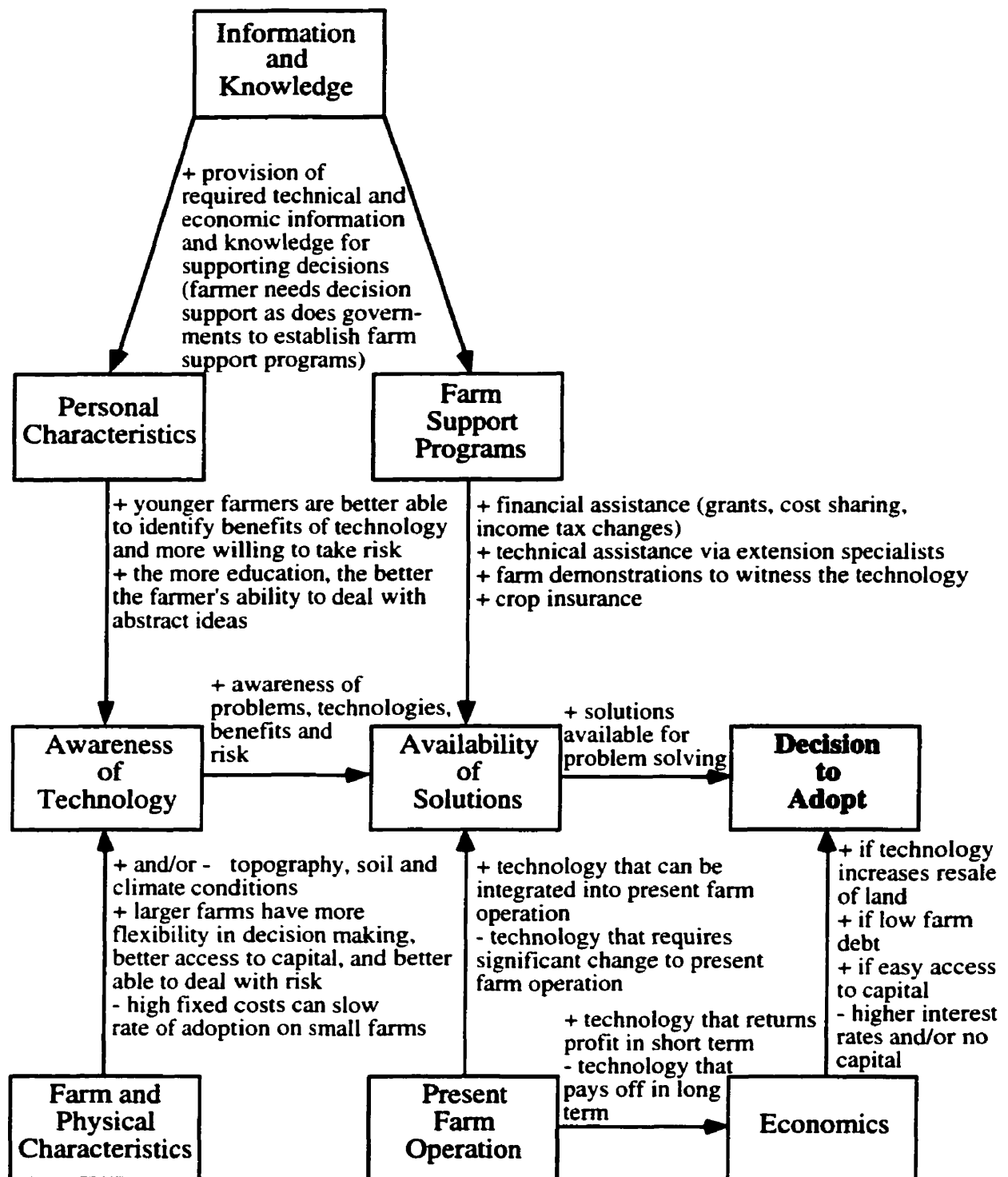


Figure 3.1: Characteristics that Influence Technology Adoption
 ("+" means positive influence, "-" means negative influence)

3.2.8 Internal Management

Internal management involves the farmer, or a farm employee, fully learning the technology(s) such that they can manage and utilize the technology successfully. The required knowledge could be acquired through on-going experience, formal training or through self taught exercises including contact with agricultural professionals and other knowledge sources (e.g. neighbors) and information.

3.2.9 External Management

Although Thomsen, in 1985, first addressed the concept of field consultants and advisors, this concept has become much more popular in recent years because of the recent growth in the size of individual farm enterprises. A variety of firms, from agricultural supply dealers to independent professional agronomists, offer farmers crop production consultation services to solve specific production problems and/or recommend complete production systems. With regards to the latter, the consultant provides a complete analysis of the present farming system, making recommendations for change related to crops to grow, which seed variety(s) to use, planting rate and date, fertilizer selection and placement, pest management strategies, and harvesting and marketing possibilities. Field consulting firms also offer monitoring services which include scouting farmer's crops for weeds, insects, disease, moisture extremes, nutrient deficiencies and other problems and recommending appropriate action. In doing so, field consultants and advisors, can assist in the adoption of complex technologies. They use some of these technologies, such as computerized decision support systems, for problem solving, which in turn are made familiar to, and adopted by farmers.

A market assessment survey [UTI, 1997a] indicates relevant findings with regards to field consultants and advisory firms, including:

- 1) The use of agriculture consultants that provide "value added" services will increase;
- 2) Early adopters of on-farm management computer software not only include individual producers but also independent consultants, company representatives, manufacturers and government agencies;

- 3) Individual producers want to collect on-farm data and will pay consultants for analysis and interpretation of their data.

3.2.10 Behavioral Influences of Technology Adoption

There has been an enormous number of agricultural innovations in the last century, of which many have been adopted and others not. With soil erosion and sustainable agriculture being one of the most important problems for many years, many innovations developed have focused on soil conservation, after a situation where the adoption of an earlier technology has caused problems. With farming being a complex business, the method in which technology is "packaged" is important to its success. Furthermore, farmers have more alternatives for managing the technologies they decide to adopt. The adoption process is complex with many interrelated factors and influences. The process can best be analyzed by examining the characteristics and behavioral influences of technology adoption. There is much literature which characterizes and explains the behavior of innovators and adopters. These behaviors are summarized in the following sections.

3.2.10.1 Psychological Influence

Pampel and van Es (1977) indicated the most important cause of innovative behavior is an intrinsic willingness to change, to experience new ideas, and hence, to adopt new practices. Innovativeness is dealt with as a psychological trait which manifests itself in all behavior, including the adoption of new farm practices. Implicit in much of the literature is that it is not important which practices are adopted but rather the orientation of the farmer toward new ideas. Rogers and Shoemaker (1968) characterized innovators as eager to experience new ideas, desire risk and hazard, and are willing to accept setbacks. They concluded that early adopters of innovations have a more favorable attitude toward change and risk, are less fatalistic, and are highly motivated toward achievement. From the viewpoint of economics, other studies have argued that achievement motivation, entrepreneurship, and willingness to change, are the primary personality traits that lead to early adoption of innovations [Rogers and Svenning, 1969; McClelland and Winter, 1969].

Since the type of practice considered for adoption is not crucial, innovative farmers will experience many new practices, with profit and environmental impact being only

secondary interests [Pampel and van Es, 1977]. In summary, judging from the innovators behavior characteristics, it is believed that the adoption of profitable commercial practices is positively related to the adoption of the other types of practices [Pampel and van Es, 1977].

3.2.10.2 Profitability Influence

A different viewpoint from that in Section 3.2.10.1, indicates that the adoption of new practices is based on the farmer's orientation toward profit, rather than the farmer's orientation toward new ideas [Pampel and van Es, 1977]. Hence, Bose (1962) and Rieck and Pulver (1962) believed that the primary goal of the adoption of innovations is to increase profit, and therefore, farmers who are most profit oriented, or in the best position for profit making [Cancian, 1967], will readily adopt new practices while less profit-oriented farmers will be more reluctant. In summary, if increased profits are the goal of adoption behavior, then early adopting farmers will adopt the most profitable innovations, regardless of the commercial or environmental nature of the practices [Pampel and van Es, 1977].

3.2.10.3 Farming Life Influence

A third viewpoint believes adoption of innovation is a consequence of orientation toward farming and farm life rather than strictly a business venture for a farm "business enterprise" [Wilkening and Johnson, 1961; Presser and Russell, 1965; Flinn and Johnson, 1974]. The business-oriented farmer is inclined to adopt practices which are suitable for his farm business and are closely involved in the agribusiness, commercial-market system [Pampel and van Es, 1977]. However, practices might be adopted which are not highly profitable. For instance, Goldstein and Eichorn (1961) found small-acreage farmers adopting equipment profitable only for large farms. The main factor is the commercial market nature of the practice, not the profitability or environmental impact¹⁴ [Pampel and van Es, 1977]. Alternatively, the less business-oriented farmer is more concerned with social responsibility and high regard for farm life. The ideal beliefs characteristic of this type of farmer include [Huth, 1957; Shepard, 1967; Flinn and Johnson, 1974]:

- 1) rural life is more satisfying than urban life;

¹⁴ This type of action is called goal displacement, whereby participating in the market system, by purchasing inputs, is an end in itself, rather than a method to achieve greater output [Pampel and van Es, 1977].

- 2) for both the moral and economic well-being of a person, the small family farm is superior to the large commercial farm;
- 3) living and working close to nature are rewarding and make one appreciative of nature.

It is believed that farmers who enjoy farming as a way of life are less likely to adopt commercial practices and more likely to readily adopt innovations which protect the rural environment [Pampel and van Es, 1977].

Chapter 4: Problem-Solving and Database Technologies for Decision Support Systems

This chapter reviews the literature pertaining to problem-solving and database technologies for decision support systems. Various methods, alternatives, and systems that researchers have studied and developed for providing decision support and problem solving are described. Database technologies for decision support systems are the focus later in this chapter. The chapter concludes that computerized decision support systems hold promise for complicated prairie farming systems and their distributed user clientele. The three main goals and objectives of the development of Farm Smart 2000 are stated.

4.1 Problem-Solving Systems

It is apparent from the previous sections that technology is difficult to transfer efficiently and effectively. Hence, as an alternative, many organizations are turning to artificial intelligence computer-based applications called expert systems and decision support systems that mimic the knowledge and reasoning process of human experts and specialists to help their clients make decisions.

This section describes artificial intelligence techniques and methods that are used to solve problems in developing computer-based decision support systems. Distributed artificial intelligence is discussed focusing on multi-agent systems in terms of interaction, communication and coordination among agents. The approaches to and characteristics of cooperative problem solving are examined leading to the determination of knowledge requirements for cooperative problem solving. Traditionally, the role of database management systems did not acknowledge the importance of rule management as a technique for implementing active databases. This section examines the problem of cooperative problem solving from a data/knowledge-base perspective and characterizes the software engineering and modeling support required to facilitate cooperative problem solving. The concepts of object-oriented databases and modeling techniques for supporting cooperative problem solving are included.

4.1.1 Terminology

An expert system is a computer program that relies on knowledge and reasoning (i.e. artificial intelligence) to be able to mimic a human expert in a specialized area. Human experts use their personal knowledge to reason in order to arrive at conclusions. Similarly, an expert system reasons and arrives at conclusions based on the artificial intelligence it possesses.

An *agent* is defined as a logically independent computational process representing a specific area of expertise, such as an expert system, developed to deliver a specific knowledge set [Bond and Gasser, 1988] within a decision support system of integrated expert systems (i.e. agents). Information represented in a modular and logically independent method is potentially *reusable* [Neches et al., 1991]. A reusable agent system is computationally analogous to a pool of human specialists. Hence, a *multi-agent system* is composed of agents selected from an existing library and integrated with no, or minimal, customized implementation [Lander, 1994].

Heterogeneity among agents can be classified in terms of knowledge and implementation. Heterogeneous agents, in terms of knowledge, are characterized by differences in declarative knowledge, solution evaluation criteria, goals, capabilities and priorities [Lander, 1994]. Heterogeneous agents, based on implementation, can be characterized by differences in architectures, algorithms, languages, inference engines or hardware requirements. These differences are sometimes classified as representation heterogeneity [Lander, 1994].

4.1.2 Cooperating Expert Systems/Agents

Large complex problem-solving systems often require the expertise of multiple specialized agents working together to produce comprehensive solutions. Each problem-solving agent is autonomous and capable of sophisticated problem solving. However, when a set of problem-solving agents is faced with a complex problem, cooperation is essential. These cooperative problem-solving activities also require extensive access to information repositories, computation, inferencing and decision-making.

The predominant properties of a cooperative problem-solving environment are [Chakravarthy et al., 1992]:

- decision making and inferencing - utilizing a set of criteria within a problem-solving agent (e.g. selecting a crop variety or crop rotation for a given year);
- computation - using a selection of systems and tools (e.g. computing the potential economic return on each crop selected);
- collaboration - where the capability of participating problem-solving agents and the dynamic development of the problem-solving strategy are inferred. That is, reasoning about problem solving and the necessary coordination by the system. (e.g. in order to determine economically viable candidate crops for this growing season, agents that specialize in crop rotation, plant disease, herbicide residue, crop residue, pests and fertility, all need to cooperate in solving this problem);
- coordination - where synchronization and execution of the plan are performed based on the causal relationships and actions performed by individual problem-solving agents (e.g. if the current sequence of crops indicates a potential for plant disease, then rule sets and priority settings guide the system to resolving plant disease before any other associated problems).

The two most important properties of a cooperative problem-solving environment are collaboration and coordination. The main function in collaboration is the solicitation of contributions from participating agents so that the problem can be solved jointly by the selected set of agents through coordination. The main function of coordination is to resolve conflicts, allocate limited resources, and search in global space, based on local information. Conceptually, collaboration can be viewed as the development of the plan, and coordination corresponds to execution of the plan, whether it be pre-defined or dynamically developed, or both. Clearly, collaboration and coordination are the two key characteristics which together result in cooperative behavior. It is imperative that the underlying computing environment be capable of providing support for each of the characteristics identified, in order to achieve cooperative problem solving.

Current literature indicates that there are basically two approaches to developing cooperative multi-agent problem-solving systems. One approach is to establish a global structure containing information specifically about each agent to use for agent coordination

and communication during the problem-solving process [Klein and Lu, 1990; Lander and Lesser, 1989; Lander et al., 1991; Werkman et al., 1990]. A contrasting approach is to develop all agents with inherent knowledge and information about all other agents such that agents interact directly with each other through a collective action where agents take action or make decisions based on influence from or knowledge of other agents [Bond and Gasser, 1988; Polat et al., 1993a; Polat et al., 1993b].

4.1.3 Distributed Artificial Intelligence

Historically, researchers divided Distributed Artificial Intelligence into two main areas [Bond and Gasser, 1988]. *Distributed Problem Solving* undertakes the work of problem solving by dividing the tasks among a number of cooperating modules that share knowledge about the problem and the developing solution [Smith and Davis, 1981]. Alternatively, research in *Multi-agent Systems* concentrates on coordinating the knowledge, goals, skills and plans of autonomous intelligent agents to facilitate problem solving through their joint intelligent behavior.

If there is cooperation among the problem-solving agents, then their coordination plan is completely accurate and well defined. That is, there is a specific protocol for communication among modules.

4.1.4 Multi-Agent Systems

Multi-agent systems not only share knowledge about the problem, but must also reason about the type of coordination required among the agents in order to solve the problem. The agents in a multi-agent system might work together toward a single global goal or toward separate individual goals that cooperatively interact to meet a common goal. As in a distributed problem-solving system, agents in a multi-agent system need to share knowledge about problems and solutions, and must also reason about the coordinating processes used among the agents in the cooperative environment. Coordination in a multi-agent system, particularly an open system [Hewitt, 1991], can be difficult, as situations can arise where there is no possibility for global control, globally consistent knowledge, globally shared goals, or global success criteria, or sometimes, not even a global representation of the system [Hewitt, 1985; Hewitt, 1988].

4.1.4.1 Methodologies for Multi-Agent Cooperative Problem Solving

In the same way that work is divided among human experts, there must be a division of labour and organization in order to distribute tasks among agents [Bond and Gasser, 1988]. In order to accomplish this distribution, tasks must be formulated and described in such a way as to facilitate their distribution [Bond and Gasser, 1988]. Complicated tasks, requiring more resources or knowledge than is held by a single agent, must be decomposed in order that they can be accomplished. Tasks must be allocated or assigned to particular agents that are capable of performing them. All these requirements are interdependent.

The concepts and methods used for task description and formulation will influence how tasks can be decomposed, and what dependencies explicitly exist among tasks. For instance, the same task described from different perspectives may necessitate different decomposition and different resources. Decomposition affects how tasks can be allocated, since the resources and knowledge of agents performing the tasks must match the task requirements [Lesser and Erman, 1980]. Alternatively, the dependence and communication among tasks can affect the way a task is decomposed. As the context of the overall system changes, decomposition decisions may need to be revised. In addition, the availability of resources may affect the choices of distribution and aggregation.

In multi-agent systems, in order to establish and maintain problem-solving alliances, agents may need to negotiate the appropriate formulation of problems and appropriate responsibility for the description, decomposition and allocation of decisions. Historically, research has focused on creating a flexible task allocation methodology, to be used after a problem has been described and decomposed, rather than automated dynamic problem formulation and description [Bond and Gasser, 1988].

Selecting the relevant problem-solving agent is the most important step in solving a sub-problem and thereby the actual problem. By storing and maintaining the capabilities of the reusable problem-solving agents in a database, the complex task of choosing the correct problem-solving agent decreases, and in simple cases, to merely querying the capability database. Any changes to the number of reusable problem-solving agents, or their capabilities, must be reflected through changes to the capability database.

After the relevant problem-solving agents are selected, the next important step is the coordination of these problem-solving agents to solve the actual problem [Chakravarthy et al., 1992]. At a minimum, coordination requires some knowledge of the actions of other problem-solving agents, and the ability to reason about the effect of their actions. In a cooperating problem-solving environment, it is unlikely that the entire plan and the interaction required among the problem-solving agents is well-defined at the onset. It is more likely that, except in simple cases, coordination required among problem-solving agents is dynamically derived, necessitating dynamic coordination.

4.1.4.2 Collaboration and Coordination Among Multiple Agents

In order to support cooperative problem solving, it is imperative that the "computing environment" be capable of providing support for collaboration and coordination. The "computing environment" could be comprised of many support mechanisms, including the operating system, providing a general problem-solving environment. In addition, artificial intelligence tools could be included such as conflict resolution and rule-based engines, case-based reasoning, and constraint negotiation which provide sophisticated cooperative problem-solving support when no specific collaboration and coordination strategy can be formulated. Also, enhanced database technology, including blackboard technology and active databases can be included in the computing environment providing cooperative problem-solving support when a specific protocol cannot be established for collaboration and coordination of agents. All of these provide sophisticated cooperative problem-solving support and dynamic coordination capabilities.

In order for several intelligent agents to combine their expertise and efforts, interaction and communication are necessary underlying concepts of multi-agent systems. Interaction in a multi-agent system, means a type of collective action in which agents take action or make decisions based on influence from the presence or knowledge of other agents [Bond and Gasser, 1988]. Unlike agents' perceptions, beliefs, and goals, which may or may not be distributed, interaction is inherently distributed and is necessary for the successful coordination of agents' actions.

A *rational action* is taken by an agent in response to goals it has, believing that the goal will be satisfied by the action [Bond and Gasser, 1988]. Communication normally

takes place whenever there is the intention to communicate, as agents know that this intention to communicate will provide additional information. It is important to note, that although agents may communicate coherently in order to structure coordination, communication between each other is not necessarily required in order to effectively coordinate interaction among agents. Agents may coordinate their actions utilizing models or metalevel information they have of each other, without interaction [Rosenschein et al., 1986; Tenney and Sandell Jr., 1981].

Durfee (1987) indicates that meta-level information should enable an agent to reason about the past, current and future actions of another agent. It is thought that agents that utilize redundant data are more efficient when they exchange metalevel information about the goals and plans of other agents, because it enables them to avoid redundant work. Metalevel information permits agents to more accurately assess the effects of communications on other agents' activities [Durfee et al., 1987a]. Metalevel communication has two advantages, fewer messages need to be exchanged reducing communication, and a decrease in the computation required to interpret messages and predict future actions in other agents. A disadvantage is that unless obsolete metalevel information is managed, it can be counterproductive.

4.1.4.3 Knowledge Requirements for Multi-Agent Collaboration

The collaborative aspect of cooperative problem-solving requires problem-solving agents to be able to reflect their role, as well as the roles of others involved in cooperative problem solving [Roda et al., 1991]. To accomplish this, it is necessary to determine the requirements for knowledge that need to be captured (about the agent's self as well as other agents) and maintained, the process of capturing the knowledge, and the use of knowledge in the cooperative problem-solving paradigm.

There is consensus in the research community regarding the types of knowledge that need to be captured, and these are listed below [Roda et al., 1991]:

- 1) Domain knowledge - facts and relationships regarding the environment in which the problem solving is occurring;
- 2) State information - information about the problem being solved with respect to self and others;

- 3) Capability knowledge - capability of problem solving and interaction (communication and coordination) of self and others;
- 4) Intentional knowledge - knowledge about what a problem-solving agent intends to do; usually described in terms of plans;
- 5) Evaluative knowledge - knowledge which permits problem-solving agents to distinguish between problem-solving agents that offer similar services. This knowledge is important and in addition to other information, might include a ... (See #6)
- 6) competence rating for skills - a measure of reliability attached to information coming from a particular problem-solving agent, or a measure of timeliness in a time-critical environment.

As a first step, the capability knowledge is captured in a capability data/knowledge-base. This database stores the capabilities of the problem-solving agents in the cooperative problem-solving environment. The problem-solving agent's capabilities are defined by means of a capability definition language which, once compiled, generates both intentional and extensional data of the capability database [Chakravarthy et al., 1992].

4.1.5 Problem of Conflict and Resolution

Although integrated artificially intelligent systems of heterogeneous reusable agents are potentially adaptable, maintainable and affordable [Lander and Lesser, 1994], the problem of conflict resolution plays a major role in the integration and formulation of communication and coordination protocols. Intelligent problem-solving agents must externalize parts of their world in order to reason about them [Bond and Gasser, 1988]. This externalization is subject to problems of abstraction and incompleteness due to the problem of attempting to fully represent an object or process. Therefore, intelligent agents need to contend with differences and uncertainty between their externalized representations and the affairs actually represented. When multiple agents are integrated they must externalize one another into representations which must then be aligned to coordinate with each other. Alignment does not mean that agents share representations but rather that the representations must allow them to act so as to fulfill their individual goals. Developing common representations, such as standards for communication protocols or information

exchange [Durfee, 1987; Durfee et al., 1987a; Durfee et al., 1987b] appear to be workable, but they depend very much on the agents' compatibility to interpret these protocols [Winograd and Flores, 1986]. Furthermore, agents may have knowledge-bases which contain differing beliefs, or they might have the same beliefs but with different confidence levels. Differences like these lead to conflicts among the agents.

Not all differences should necessarily cause conflict [Bond and Gasser, 1988]. Agents need to have different knowledge and structure to avoid duplication of work. If agents have redundant or overlapping data, they must be able to determine which data are relevant to them and which to another agent, to again avoid duplication. As for reasoning about differences in belief among agents, it is necessary to be able to represent other agent's beliefs [Rosenschein, 1987]. In essence, it is important to understand each agent's identity and manage their differences appropriately.

Not all conflicts can be resolved, particularly in open systems [Hewitt, 1991]. To resolve conflicts, agents must have some basis on which to agree. This basis is often established using a central or global controller with decision-making authority. Agents can also resolve conflicts by relaxing constraints or by redefining a problem to eliminate them [Goldstein, 1985]. If it is thought that conflict arose from inconsistent assumptions, backtracking is sometimes used to discover if the roots of the disparity lie in the assumptions [Sycara, 1985]. Differences can then be resolved at any point in the backtracking procedure. This approach necessitates that an agent has knowledge of the supporting assumptions and their relationships. This might possibly be accomplished by incorporating an Assumption-Based Truth Maintenance System which can represent all the relationships that have ever been considered at the same time and thereby check which assumptions might cause the discrepancy.

Human experts cooperating on a single problem contribute their multiple and diverse viewpoints during the problem-solving process. Merging diverse knowledge is common in real-world situations bringing the robustness and balance required to develop comprehensive and credible solutions. For instance, a plant breeder and an agronomist work together to develop crops with high quality and good yields for a particular growing environment. Similarly, a chemist, a machinery engineer, and a ecotoxicologist cooperate

to produce effective and environmentally safe methods of applying herbicides. In general, the greater the complexity of problem-solving, the more cooperating experts needed. Cooperation and diversity can be advantageous, providing increased problem-solving capabilities beyond the individual expert and promoting creativity and innovation [Lander and Lesser, 1989a]. However, conflicts result from cooperation which must be resolved among the experts through information exchange and problem-process control.

4.1.5.1 Solutions for Conflict Management

There are several existing approaches to conflict management within the paradigm of cooperating expert systems. Klein and Lu (1990) managed conflict by establishing an abstraction hierarchy of conflict classes each with an associated applicable conflict resolution strategy. When conflict is detected, a conflict resolution expert uses the global conflict hierarchy to resolve it. Similarly, Werkman et al. (1990) within their Design Fabricator Interpreter system for distributed cooperative problem solving among construction agents, utilized a third-party arbitrator agent to resolve conflicting recommendations based on global conflict resolution knowledge. Sycara (1993) addressed conflict resolution by incorporating a negotiation model, which integrates case-based reasoning. Case-based reasoning retains the process and the results of its decision, and thereby can shorten similar reasoning chains by referring to the stored process and results as well as recognizing and avoiding similar failures in advance.

Lander and Lesser (1989a) developed several methods for managing conflict. One method is based on a generic cooperating experts framework to support cooperative problem solving among sets of knowledge-based systems [Lander and Lesser, 1989a; Lander and Lesser, 1989b]. Within this framework, conflicts are resolved by employing compromise and integrative negotiation. Compromise negotiation involves a concession of requirements on the part of each agent in order to satisfy other agents' requirements. Integrative negotiation identifies the most important goals of each party to use in formulating a basis for a new solution. In an extended variation of the same cooperating experts framework, Lander et al. (1991) utilize conflict resolution strategies that are appropriate to the problem-solving context [Lander et al., 1991]. Lander and Lesser (1992) also developed a negotiated search algorithm which can be used in different

variations of conflict management [Lander and Lesser, 1992]. Finally, Lander and Lesser (1994) in a different approach, used meta-information to resolve conflict and improve joint results [Lander and Lesser, 1994]. Each reusable agent shared meta-information with other agents, assimilated shared information from other agents, and then used this information to improve their local understanding of the global solution space to achieve global solutions.

In problem solving that includes heterogeneous and reusable agents, each agent can determine the status of its local solution, but there is no effective method for an agent to determine the status of the global solution, as no agent has knowledge of the constraints of other agents [Lander and Lesser, 1992]. Although researchers [Sycara et al., 1991] have addressed this problem to some extent in their literature on constraint-directed negotiation, the investigations they conducted on this subject in a multi-agent environment necessitated that the agents share a globally consistent problem-solving methodology and agent architecture. These agents were heterogeneous, based on their resource requirements, but homogeneous in their implementation, and implicitly used their homogeneity for control.

Although conflict resolution can be attempted using any of several methods of negotiation, compromise negotiation holds the most promise for dealing with negotiation among heterogeneous reusable agents. Compromise negotiation uses a methodology where values are iteratively revised by sliding them along some dimension until a mutually agreeable position is found [Lander and Lesser, 1989b]. An example of compromise negotiation is demonstrated when a customer is purchasing a car. The car buyer and salesman iteratively propose prices (i.e. slide a value along a monetary scale) until the two proposals converge. For compromise negotiation to be effective, a small number of dimensions should be involved, a method should exist for determining whether the proposed values are moving toward each other, and the values should be within an acceptable range [Lander and Lesser, 1989a].

An alternate approach to conflict resolution among heterogeneous reusable agents is relaxation. Relaxation methods are invoked when the problem-solving activity is believed to be over constrained [Lander and Lesser, 1992] resulting in no or little progress in solution building. Relaxation is invoked when one or more agents relax some requirement on a goal or solution, thereby enabling problem solving to advance. Sometimes through

relaxation, a case that had failed previously may be stored in the case history database and becomes immediately acceptable. Within each agent's meta-knowledge, weighted constraints are assigned to each information type or variable, which define the degree to which that agent is willing to relax that information [Lander and Lesser, 1992].

Klein and Lu's (1990) run-time model was oriented towards cooperative conflict resolution based on the notions that conflict-resolution expertise can be captured explicitly and organized usefully into a taxonomy of conflict classes each with an associated and applicable strategy for conflict resolution. They supported and instantiated this model in a study of conflict resolution among human experts in the domain of architectural design. In contrast to competitive conflict situations, where each agent has solely their own interest and benefits in mind, cooperative conflict resolution strategy strives to achieve a globally optimal solution, sacrificing individual goals in the interest of increased global benefit. Hiding information from the adversary and making threats are common strategies in competitive conflict situations. Whereas strategies for cooperative conflict situations usually include compromise or surrendering less important goals in order to reach as mutually beneficial solution as possible.

In Klein and Lu's (1990) abstraction hierarchy of conflict classes, the more general conflict classes were arranged near the top of the hierarchy and represent domain independent classes, whereas the more specific classes were near the bottom and relate to domain dependent classes. Conflict resolution strategies within the domain dependent classes will have a narrower range of applicability, but usually greater efficiency, than the general strategies contained in the domain independent classes. When a conflict occurs, this hierarchical structure facilitates the search for the most specific conflict class matching the conflict, and tries the conflict resolution strategies associated with that class. If these strategies are unsuccessful, progression can proceed up the hierarchy to the next conflict class where the more general and less efficient strategies associated with that class are tried. Hence, if no specific conflict class for a particular conflict was encoded, the hierarchy can be searched to locate the closest conflict class associated with that kind of conflict. There is a trade off between the main two segments of the hierarchy with the domain independent segment being more applicable but less efficient and the domain dependent segment being

more efficient and less generally applicable. General conflict classes would be difficult and time consuming to apply, and hence, Klein and Lu (1990) recommend developing enough specific conflict classes and strategies to avoid the use of these expensive general strategies.

Although the domain independent levels of the conflict hierarchy are generated once during the development of the design system, based on acquired knowledge from domain experts, the domain dependent levels need to be recreated for each substantially new domain. Within Klein and Lu's (1990) cooperative architectural design system, the conflict hierarchy worked as follows. All design agents cooperated with each other by contributing to a shared representation of the design and criticizing other design agents contributions. A conflict resulted when two incompatible design contributions existed or when an agent's commitment was criticized. A description of the conflict was then communicated to the conflict resolution expert which utilized the conflict hierarchy to determine a solution for the conflict.

In a similar approach, Werkman et al. (1990) within their Design Fabricator Interpreter system, utilized a third-party arbitrator agent to resolve conflicting recommendations based on global conflict resolution knowledge. The Design Fabricator Interpreter was a system for distributed cooperative problem-solving between construction agents. This system resembled the distributed nature of the construction industry by incorporating a multi-agent architecture which models design, fabrication and erection processes [Werkman et al., 1990]. The architecture, by using negotiation, considered each participant's important issues in the connection design process and produced a cooperative solution. The representation of specialized construction process knowledge in each agent's framework of models, facilitated the addition of new knowledge as it was acquired. The modular nature of the architecture enabled new agents with their new construction expertise to be added with little customization required.

The Design Fabricator Interpreter system evaluated multi-agent viewpoints and suggested alternative connection designs. During the proposal process, the agents formed their viewpoints with both cooperative and competitive behavior. They worked jointly toward a common goal by suggesting alternative configuration connections which resembled the connection design originally specified by the user, but simultaneously,

competed with each other, attempting to maximize their own positions within the problem-solving domain. An agent formulated its viewpoint from several unique issues based on different connection aspects such as economics, feasibility and material type. Within a connection evaluation process, the importance of a particular agent issue is determined by which agent viewpoint was taken, designer, fabricator or erector. When making a proposal, an agent would evaluate the connections previously proposed by other agents for any affected issues. During the agents' proposal process, a third-party arbitrator monitored the proposals and mediated by accessing an abstract level shared knowledge-base containing agent's issues. The negotiation process produced a final set of alternative connections which was usually acceptable to all agents.

Sycara (1993) uses case-based reasoning extensively in the development of the PERSUADER, a decision support system which provides enhanced conflict resolution and negotiation support for group problem solving, specifically labour management disputes. Case-based reasoning is an artificial intelligent approach to machine learning where the process and results of decisions (i.e. cases) are retained and subsequently used to solve future related problems (i.e. case-based reasoning). As experiences and sessions are accumulated, system performance improves providing more intelligent support, and in essence, the system learns from experience. There are several advantages of using case-based reasoning as a learning methodology [Sycara, 1993]. Incremental learning can be pursued without attention to the order of presented cases. Previous successful decisions can be exploited providing short cuts to current problem solving. Similarly, previous failures and reasons for failure can be used to recognize and avoid similar failures in advance, as well as support recovery from current problem-solving failures. If the repair of a failure is also stored in the database, then the repair can be accessed and re-used for a similar problem. At the conclusion of each problem-solving session, the case database is updated with the process and results of the decision-making session.

In the PERSUADER, a case was solved by first accessing relevant precedent cases from the case database and selecting the most appropriate cases. Using the most important features from the current case, a baseline solution was constructed which was adapted to the details of the current problem using heuristics and case-based reasoning. The baseline

solution was then evaluated for applicability to the current case and modified accordingly to obtain a candidate solution.

Within the case database, cases were structured hierarchically based on important concepts in the problem domain. Cases were retrieved based on whether they shared salient attributes with the current case. Sycara (1993) used a high-level knowledge structure called generalized episode [Kolodner, 1984] to organize similar concepts in the case database. Concepts within a generalized episode structure were organized in a hierarchical discrimination network where nodes were either an individual case or another generalized episode [Sycara, 1993]. Within a generalized episode, there were norms which were collections of features that represent the abstraction of all the cases which were organized under this generalized episode. Indices were used to connect a generalized episode with other generalized episode trees and cases organized below it. In the generalized episode, cases were indexed based on the features which differentiate them from the norm. As cases were introduced and added to the case database, the generalized information describing the collection of features in the norm part of an existing generalized episode were improved and additional case indices were added. In this method, salient attributes were learned and incorporated into the case database structure [Sycara, 1993].

4.2 Database Technologies for Decision Support Systems

Traditionally, the role of database management systems has been to provide shared access to large volumes of data, and to support consistency, recovery and other basic aspects of sharing [Chakravarthy et al., 1992]. The roles of applications are embedded in programs that execute over the database. However, recently there has been a shift in the way databases are used, and the support and functionality expected from them. This shift has come about as a result of the recognition for the need and utility of databases for non-traditional applications. Furthermore, database management systems have evolved by internalizing additional functionality to fulfill the requirements of new types of applications.

Most applications of distributed artificial intelligence [Bond and Gasser, 1988] use long term memory to refer to the database and short term memory to refer to the active blackboard. Bond and Gasser (1988), address the collaboration and coordination aspects by means of goal setting and processing, and treat the long and short term memory as

second class objects. In contrast, this section discusses the extended capabilities of a database management system to develop a database supported approach to cooperative problem solving.

An important notion with regards to cooperative problem solving is that of problem-solving agent's capability modeling, that is, how to model oneself, in terms of domain knowledge, capability, intentional knowledge, and evaluative knowledge, in conjunction with other participating agents. Another important aspect of cooperative problem solving is the representation of domain knowledge and methodologies for their use in problem solving. Other aspects/problems to be considered in a cooperative problem-solving environment include [Sheng and Wei, 1992]:

- coherence - the overall behavior of the system as a unit along some dimension of evaluation;
- efficiency - the overall efficiency of the system in solving a problem;
- uncertainty - having incomplete or inaccurate information about aspects of the problem-solving (e.g. about the state, action, or plan).

4.2.1 Conventional Database Management Systems

There are some fundamental differences between conventional or traditional databases and cooperative problem-solving database systems [Chakravarthy et al., 1992]. Traditional databases have endeavored to capture mostly the static description of data, by means of a logical data model, to be managed and maintained by the system, whereas in cooperative problem-solving database systems, the data/knowledge-base changes frequently. Secondly, the transaction model used in traditional database management systems is based on a paradigm that views concurrent transactions as "competing" for resources rather than "cooperating" to solve a specific problem. Lastly, in conventional database systems, coordination among constituent nodes is pre-defined and the problem decomposition uses a known coordination structure, whereas in the cooperative problem-solving paradigm coordination is dynamically determined.

One approach to achieve dynamic coordination within database systems is to generate "activity" rules that can be used for dynamic coordination of activities. These rules are more specifically termed event-condition-action rules [Diaz et al., 1991]. This

dynamic coordination of activities using event-condition-action rules is often referred to as active databases.

As the focus of this section is on database support for cooperative problem solving, a review of traditional databases, in terms of their support for collaboration and coordination, is beneficial to the discussion of enhanced database support to follow.

4.2.2 Distributed Database Systems

Distributed database systems have very little or no collaboration and all the coordination is known *a priori* [Chakravarthy et al., 1992]. Distributed database system support in cooperative problem-solving environments is reduced to that of populating and maintaining a data dictionary containing extensional knowledge and system capabilities which are stored in each node. Much of the coordination is formulated by the distributed query processor and the distributed transaction management subsystem by using the information maintained as part of the data dictionary. Hence, collaboration is reduced to data exchange, and coordination is accomplished by the system components using commit protocol and other techniques. Furthermore, a problem within a distributed database system must be decomposed and individual subproblems are sent to various problem-solving agents. The problem-solving agents are assumed to be homogeneous and possess the same functional capability.

4.2.3 Heterogeneous/Multi-Databases

Heterogeneous databases, also termed federated or multi-databases, are similar to distributed database systems with some exceptions [Chakravarthy et al., 1992]. In heterogeneous database systems, there is no explicit collaboration but the coordination takes place over heterogeneous problem-solving agents instead of homogeneous agents, and as a result, problem-solving agents need to be modeled in more detail. The coordination is still system oriented but the commit protocol is more complex requiring, in some cases, human interaction. The differences between the problem-solving agents, and their capability in terms of functionality, are captured as part of the data dictionary, or global data dictionary.

4.2.4 Database and Knowledge-Base Systems

Database and knowledge-base technologies are often complementary rather than contradictory. Normally, a database maintains well-structured data representing the facts that traditionally are essential to data sharing and processing activities, whereas a knowledge-base is mainly for decision and planning support, and hence contains less precise, more abstract, and possibly subjective knowledge. Coupling knowledge-base and database technologies improves data management by using knowledge-base techniques to manage complex relationships among data, as well as to perform deductive data processing. Furthermore, the merging of these technologies improves knowledge management by using database techniques to maintain the factual data imbedded in knowledge, thus reducing the size and improving the extensibility and maintainability of knowledge-bases.

In order to efficiently design the intimate data and knowledge inferencing operations required in a coupled knowledge-base/database system, it is critical to properly represent the related intensional knowledge including the structural knowledge (i.e. data semantics involved), the general procedural knowledge (i.e. data processing procedures), the heuristic knowledge (i.e. the problem-solving heuristic rules in the application domain), and the control knowledge (i.e. the knowledge processing procedures). Of these four forms of knowledge, control knowledge poses a problem as this knowledge has traditionally been imbedded in the knowledge-base implementation environment, for example, in the forward-chaining method of an expert system shell. Implementation of coupled knowledge-base/database systems will require that this knowledge be developed and included in the design methodology. These four forms of knowledge are closely related rather than independent. The control knowledge uses the heuristic, the general procedural, and the structural knowledge. The heuristic knowledge uses the general procedural knowledge. The general procedural knowledge uses the structural knowledge. Therefore, the conceptual knowledge-base/database model needs to effectively support extrapolation of high-level knowledge from low-level knowledge, as well as access low-level knowledge during the inferencing of high-level knowledge. Based on these requirements, Sheng and Wei (1992) present a synthesized object-oriented entity-

relationship model that synthesizes and extends the concepts and the notations belonging to the families of object-oriented and entity-relationship models. Object-oriented modeling concepts represent the structural knowledge using object classes and their relationships, and encapsulate the general procedural, the heuristic and the control knowledge with the appropriate object classes involved. Furthermore, the organization of the general procedural and the control knowledge conforms with the underlying structural knowledge.

Structural knowledge includes information about entities and their associated relationships. Constructs for the structural knowledge can be developed from object-oriented concepts. Objects and classes, consisting of attributes and operations, can represent the structural entity knowledge, whereas the structural relationship knowledge can be represented using the three object-oriented relationships; aggregation, association and generalization.

General procedural knowledge is similar to knowledge about appropriate procedures on a given data set. Procedures are used to represent the general procedural knowledge which refers to data operations performed on classes. Procedures are encapsulated within classes and hence procedures of a superclass are inherited by its subclasses.

Heuristic knowledge is knowledge concerning the expertise of domain experts. Decomposition of heuristic knowledge according to levels of expertise into sub-categories, such as application-specific, focus management and enterprise-directing knowledge, can be incorporated within this knowledge taxonomy. Rules are used to represent the heuristic knowledge. Each class has a set of rules which describe the heuristic pertaining to that class. Like operations of a class, rules can be applied to objects in the class. Through the object-oriented inheritance concept, the rules in a class can be inherited by its subclasses. This should provide a method for structuring rules within class hierarchies.

Control knowledge determines the process of heuristic knowledge reasoning. An example could be ensuring that the heuristic knowledge imbedded in a subclass object will be inferenced before the inferencing of the heuristic knowledge related to other associated objects. The assumption that coupled knowledge-base/database systems will be implemented in an object-oriented environment will require that this knowledge be

developed and included in the design. Since the heuristic knowledge in an object-oriented approach is structured using class hierarchies, the corresponding control knowledge includes searching within class hierarchies, dynamically creating required sets of objects, processing heuristic knowledge, that is rules, of all created objects, and then deleting all these objects. A description of the creation and deletion of these domain-independent objects follows.

Classes can own triggers, for the purposes of searching within class hierarchies and dynamically creating objects. Searching within one class hierarchy is enabled by the trigger of a superclass that determines an appropriate subclass through the generalization relationship and activates the appropriate subclass's trigger. This results in making the heuristic knowledge of a more specialized class accessible. Whether association or aggregation relationships exist between two class hierarchies will govern whether searching from one class hierarchy to another is allowed. This type of searching could be accomplished by having a trigger in one class hierarchy activate a trigger in another class hierarchy. Objects can be dynamically created by triggers during these search processes. Hence, triggers can be treated as a dynamic object creation mechanism. By dynamically creating objects, heuristic knowledge of the required objects can be derived and then be applied.

As a first step, in developing an approach to cooperative problem solving among reusable agents utilizing enhanced database technology, capability knowledge (i.e. agent meta-knowledge) must be captured in a capability data/knowledge-base. This database stores the capabilities of the heterogeneous reusable problem-solving agents as they relate to the problem-solving environment. The problem-solving agent's capabilities are defined by means of a capability definition language which, once compiled, generates both intensional and extensional data of the capability database [Chakravarthy et al., 1992].

4.2.5 Blackboard Model

Blackboard systems are a tool or framework for developing cooperative problem-solving systems. Corkill (1991) uses a "blackboard" metaphor to introduce and explain blackboard-based problem solving, although, in 1962, Newell is noted as being first to use the term "blackboard" in the artificial intelligent literature and used a similar metaphor.

In this metaphor, it is envisioned that a group of human specialists are gathered around a large blackboard to cooperatively work together to solve a problem, similar to the agents illustrated in Figure 4.2. The problem-solving session begins when the initial data and problem are described on the blackboard. The specialists (i.e. agents) examine the information on the blackboard and continually look for an opportunity to apply their expertise to the developing solution. When sufficient information is available for any of the specialists to make a contribution, the contribution is recorded on the blackboard. This now allows other specialists to apply their expertise. This process of adding contributions and information to the blackboard database continues until the problem is solved. This metaphor identifies several important blackboard-system characteristics which are discussed below.

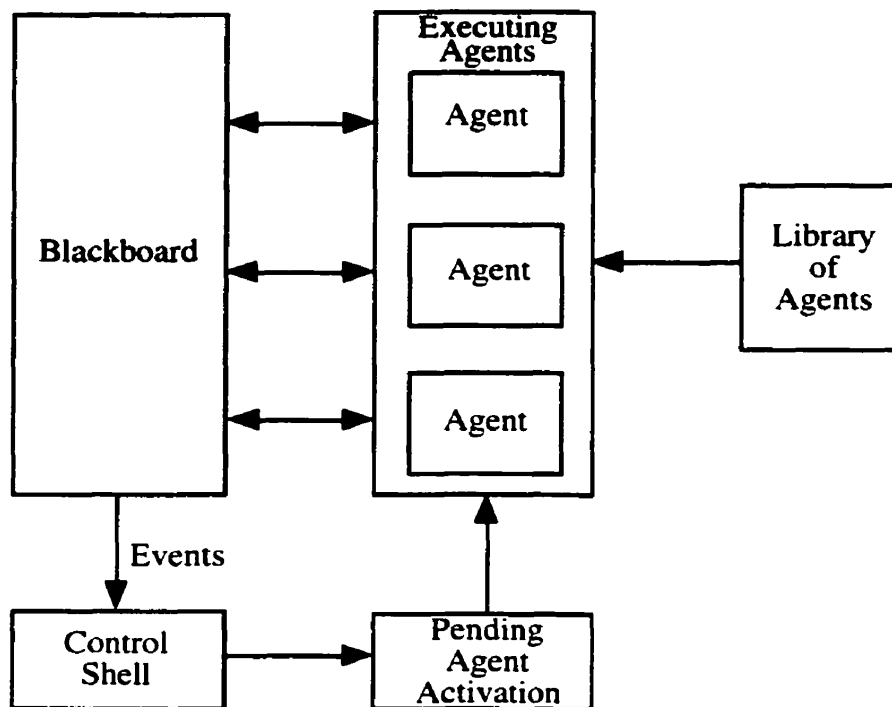


Figure 4.2: Blackboard-Based Problem-Solving Concept

In computational terms, the problem solving is performed by cooperating knowledge sources (e.g. agents or expert systems) which contain the knowledge required to solve the problems and are kept separate and independent. Various knowledge sources can be developed for any one application and each can use whatever technology best matches its purpose. The knowledge sources interact anonymously using a shared

structured global database (i.e. the blackboard). Each knowledge source has continual access to the state of the solutions on the blackboard and therefore can contribute opportunistically. Hence, the most appropriate and effective knowledge is applied to the developing solution at the best time. The problem solving is directed by a control shell that is separate from the knowledge sources. Events (i.e. signals) form the communication between the control shell and the blackboard. Events trigger knowledge sources.

The blackboard model is a relatively complex problem-solving model providing the conceptual organization of information and knowledge, and prescribing the dynamic control and use of it pertaining to the problem-solving behavior within the overall system [Engelmore and Morgan, 1988; Jagannathan et al., 1989]. The blackboard data structure, in its most general form, consists of a global data area containing problem-solving state data. Agents incrementally make changes to the blackboard based on their problem-solving ability at a given time. The agents communicate and interact solely through the blackboard. A controlling mechanism (e.g. control shell), is responsible for determining which agent to activate, when, and using which part of the blackboard. In addition to organizing information and knowledge, blackboard systems have a particular reasoning behavior associated with them as a result of incremental problem-solving. At each solution step or control cycle, any type of reasoning step can be initiated in response to the part of the emerging solution being addressed. Hence, the control mechanism selects and applies agents dynamically and opportunistically rather than by fixed and preprogrammed methods. In some cases, the control mechanism might activate an *a priori* determined set of knowledgeable sources. Although in this discussion, the blackboard model is used to integrate agents (i.e. expert systems) it is also a popular structuring architecture for building expert systems.

An expert system is built by a similar method, as a collection of interacting knowledge sources that have been developed to work together to solve a common problem. The literature indicates that several expert systems have been developed based on the blackboard model and yet they vary significantly within the blackboard framework, demonstrating the utility and flexibility of the paradigm. The literature suggests that the blackboard model is particularly suitable for systems representing multiple areas of

expertise and for systems solving problems with complex information interdependencies [Ensor and Gabbe, 1988].

The blackboard paradigm, although relatively simple to describe, is deceptively difficult to implement effectively for a particular application [Corkill et al., 1988]. Nii (1986) indicates that the blackboard model, with its knowledge sources, shared blackboard database, and control mechanism, does not specify a methodology for designing and implementing a blackboard system for a particular application, but rather is only a conceptual model.

4.2.5.1 Independence of Expertise

The specialists in the metaphor were not trained and educated to work with one specific group of specialists, but rather, they developed their expertise in different experiences. Each specialist or agent has specific expertise relating to the problem and can contribute to the solution totally independent of the other specialists. Similarly the software modules (i.e. agents) surrounding the blackboard system in Figure 4.2 are, for the majority, unrelated, each developed to exhibit expertise on one particular facet. This capability enhances the opportunity for creating an open system as inappropriate agents can be removed from the blackboard system, new agents can be added, and poor performing agents can be updated, without regard to the other agents.

If we describe the agents in Figure 4.2 as knowledge sources, then we can say that each knowledge source is like a specialist at solving certain aspects of the overall problem. Once a knowledge source finds the necessary information on the blackboard, it proceeds without any assistance from any other knowledge source to make its contribution to the blackboard.

Rule-based type systems are also modular, but only at the individual rule level. Hence, the small size of each rule prohibits full independence, unlike the large-grained scope of knowledge sources. For example, two rules which initiate iteration by one rule using a counter value and the other rule being used for termination, cannot be deleted individually without effecting the other, and nor can they be designed to work independently.

4.2.5.2 Diversity in Problem Solving Techniques

Human experts are not alike, they think and solve problems in different ways and yet this does not prevent the group of specialists in the blackboard metaphor from solving the problem. With the blackboard approach, each knowledge source (e.g. agent) is viewed as a black box in which its internal workings are invisible from the outside. It has no effect if one agent is a rule-based type expert system and another is a model. Each agent, with their individual approach, can make its contributions within the blackboard framework. Similarly, human experts are not alike, they think and solve problems in different ways and yet this does not prevent the group of specialists from solving the problem.

4.2.5.3 Flexible Representation of Blackboard Information

In the metaphor, the specialists could use various intelligible means of contributing to the blackboard. For instance, they might use algorithms, formulas, rules, diagrams, sentences, checklists, and circles and arrows. The flexibility of representation is similarly important in blackboard systems. The blackboard approach does not define any prior restrictions on the type of information that can be contributed to the blackboard.

4.2.5.4 Common Interaction Language

Although flexible representation of information on the blackboard is important, it is equally important that this information be commonly understood by all specialists in order for them to interact. If the group of specialists was of different nationalities and used different foreign languages in their sentences, formulas, diagrams and checklists, the absence of a common language would hamper or even prohibit sufficient interaction to solve the problem. Similarly, with blackboard systems, agents must be able to correctly understand the information placed on the blackboard by other agents.

4.2.5.5 Event-Based Activation

In the metaphor, the specialists do not interact directly with each other, but rather they watch the blackboard for changes which in turn give them the opportunity to act. In addition, specialists might react to external events such as noticing it is time for lunch or receiving a telephone call. Agents in the blackboard model can be activated in the same method in response to changes on the blackboard or to external events. In the metaphor, each specialist scans the blackboard for opportunities. Alternatively, each agent can

describe to the blackboard system the kinds of events it is interested in addressing, which in turn considers triggering the agent whenever that kind of event occurs. For example, a herbicide expert system planner might inform the blackboard system to trigger it whenever a question arises concerning herbicides.

4.2.5.6 Need for Control

Rather than having many specialists rushing to the blackboard in response to the same event, it would be more efficient and effective if there was an appointed manager, apart from the specialists, who would consider each specialist's request to approach the blackboard. In determining which specialist goes to the blackboard next, the manager would consider what the specialist can contribute and the effect that particular contribution might have on the developing solution.

Blackboard systems have a similar control component, separate from the individual agents, which is responsible for managing the course of action for the problem-solving process. The control component is like a specialist, directing the problem solving by considering the overall benefit that would be achieved by activating agents. In order to make and optimize these control decisions, the control component receives an estimate from the triggered agent regarding the quality, importance of its contribution, and its associated cost (i.e. resources required) if executed. The control component analyzes this information in determining how to proceed.

4.2.5.7 Incremental Solution Generation

In the metaphor, the solution is built incrementally with each specialist adding contributions, and refining and extending others contributions. A blackboard system operates incrementally as well with agents appropriately contributing, changing and initiating new lines of reasoning.

4.2.5.8 Summary

Blackboard systems are not new in the field of artificial intelligence. As indicated at the beginning of this section, Newell (1962) first made reference to the blackboard approach. The Hearsay-II speech-understanding system, developed in the early 1970s, was the first blackboard system. In the latter part of the 1970s many research projects were initiated to develop blackboard applications including HASP and DENDRAL, both of

which were developed for signal interpretation problems. HASP and DENDRAL were "expert systems", although the term had not been coined yet.

Since these early systems, the application areas for blackboard systems has broadened considerably and includes, for example, process control, planning and scheduling and, command and control. In addition, knowledge-based simulation and instruction, and case-base reasoning are also areas of application.

In the past ten years, tool kits and frameworks have been developed to aid in the increasing development of blackboard system applications. Two such blackboard frameworks are the *BB1*TM System by Barbara Hayes-Roth from Stanford University. The second is the *GBB*TM framework, originally from University of Massachusetts at Amherst, but now commercially available from Blackboard Technology Group, Inc.

4.2.6 Blackboard-Based Databases

The blackboard database, a special form of object-oriented database, is specifically designed to provide efficient management of global information. It provides a shared repository for various types of global information required in an environment and includes for example, problem specifications, partial and completed solutions, suggestions about how to proceed with a problem, information about the environment itself, and evaluations of completed and in-progress work [Blackboard Technology Group, Inc.]. This globally shared information repository provides the functionality for the collaboration aspect of cooperative problem solving, by enabling problem-solving agents to reflect their role, as well as the roles of others involved in cooperative problem solving. This role is accomplished by specific types of knowledge being captured and maintained to support agent collaboration.

The blackboard database is developed and optimized to be heavily used and highly dynamic with access and retrieval mechanisms designed for efficiency. Because it is globally visible, agents are able to prevent replication of locally shared information, avoiding not only the costs associated with duplicate data storage, but also problems that arise from maintaining currency in local copies of shared data. Information stored on the blackboard database must be represented in a shared language which is domain-dependent.

Blackboard tool kits are available providing a full range of object-oriented capabilities with which to specify the shared language.

Events provide the actions associated with data creation, modification, and access which in turn can be used for control processing. Extremely flexible control can be achieved by abstracting shared information away from the agents that provide and use the data. For example, if an agent that supplies a particular output is replaced by an agent providing the same output, the change is completely transparent to agents using that output. That is, nothing needs to be modified to propagate the replacement through the environment [Lander et al., 1996]. However, even with the benefits of the blackboard database, some developers believe it is sometimes more reasonable for agents to share large blocks of information directly with each other rather than through a blackboard. For example, legacy and "off-the-shelf" software often require their inputs to be available in specific, formatted input files, and in turn, write their outputs to specific, formatted output files. In these cases, a second level of sharing is supported by the blackboard architecture. Instead of placing all the information contained in the files on the blackboard, an object can be placed on the blackboard that describes the data and its format and points to the appropriate files. The creation of this object ensures that all agents know the data exists and acts as a triggering event for process control. [Lander et al., 1996].

Relational databases are built on the traditional concept of the "table". Consequently, a relational database management system is developed to manage tables, and in so doing, consists of three major parts: 1) data that is presented as tables, 2) operators for manipulating tables, and 3) integrity rules on tables. A blackboard-based database is a very specialized form of object-oriented database which is built on "objects", forming the most significant difference between blackboard-based databases and relational databases. The blackboard database's special object-oriented database features give it a combination of capabilities with power for expressiveness. A typical blackboard system application might contain many thousands of blackboard objects of which object insertion and retrieval is most important, in order to provide an efficient global database system. A pattern matching methodology is used for retrieving simple and composite objects from the blackboard database. An application programmer has the capability to insert additional procedural

filtering functions into the basic retrieval process. This can be significantly more efficient than applying the filters to the results of the retrieval (a method used in relational database technology).

A blackboard-based database is a global database containing objects from a solution space and can include input data, partial solutions, and other data in various problem-solving states, that are needed by and produced by knowledge sources. The knowledge sources use the blackboard data (i.e. objects) to indirectly interact with each other. The objects in the blackboard database are hierarchically organized into levels of analysis. The properties (i.e. information) associated with the objects on one level serve as input to a set of knowledge sources, which then place new information on the same or other levels. These objects and their properties define the vocabulary of the solution space with the properties being represented as attribute-value pairs. Each level in the hierarchy uses a distinct subset of the vocabulary. Often, the names of the attributes on different levels are the same. For example, "type" might often be used as a shorthand notation for "type-of-x-object" or "type-of-y-object". Furthermore, sometimes for convenience, there are duplicates of the same attribute. Named links provide the relationships (i.e. special kinds of properties) between the objects. These relationships can exist between objects on different levels, such as "part-of" or "in-support-of", or between objects on the same level, such as "next-to" or "follows". A blackboard can have multiple blackboard panels. That is, a solution space can be partitioned into multiple hierarchies.

In a blackboard system, there is a distinct separation between the database-support subsystem and the control level which allows different control shells to be implemented using the common database-supported subsystem. The interface between the two subsystems is a set of blackboard events which specify the creation, modification or deletion of blackboard objects. Relational database technology employs a very rigid command structure for data manipulation and queries.

Naturally, the notion of objects is important to the overall methodology of a blackboard system and therefore extensibility of the blackboard database is equally important. The difference between extensible conventional database management systems and object-oriented database management systems is that the former provides physical or

architectural extensibility whereas the latter provides logical extensibility, that is, the ability to define new types of data and operations on them which is very important to a blackboard-based database.

The current technology by which blackboard-based databases are implemented is object-oriented database technology and extensions thereof. In the implementation of blackboard-based databases using object-oriented database technology, knowledge sources not only insert new hypotheses (in the form of objects) in the database, but also perform associative retrieval to locate relevant hypotheses placed on the blackboard by other knowledge sources. A knowledge source is usually invoked by one or more stimulus objects. Once invoked, the knowledge source searches the blackboard to locate other objects that are appropriately related to the stimulus object. Hence, each knowledge source spends time:

- 1) retrieving objects in the blackboard according to their location;
- 2) performing computations using existing objects in determining which new blackboard objects to create;
- 3) creating and placing these new objects in the blackboard database.

The ratio of points 1) and 3) over point 2) identifies the amount of time the knowledge source spends interacting with the blackboard versus the amount of time the knowledge source spends performing problem-solving computations. The larger this interaction/computation ratio, the greater blackboard database efficiency issues will dominate overall performance of the blackboard system, and herein is the main limitation of using object-oriented database technology as the framework for blackboard databases. This is a non-trivial concern as associative retrieval is central to the blackboard paradigm. Associative retrieval provides anonymous communication among knowledge sources by allowing knowledge sources to search for relevant information (i.e. properties) in the blackboard database rather than receiving the information via direct invocation by other knowledge sources. This anonymous communication is the main underlying methodology of blackboard system cooperative problem solving.

The amount of time a knowledge source consumes creating and searching for objects compared to performing problem-solving computations (i.e. the

interaction/computation ratio) can vary greatly between applications, and even between different knowledge sources in the same application. Naturally, the greater this ratio, the more concern for efficiency of the blackboard implementation. Some developers have minimized this efficiency concern by enhancing object-oriented database technology to operate at memory rather than disk (i.e. file) speed. However, even with caching, the throughput issues of disk-based object-oriented databases can cause performance concerns.

4.2.6.1 The Blackboard Control Shell

As in our blackboard metaphor, where a manager is required to prevent eager human specialists from trampling each other in their rush to obtain the chalk, agents need a mechanism to organize their use in the most effective and coherent method. Blackboard system control has traditionally been opportunistic or reactive as in the blackboard metaphor. That is, agents react to changes on the blackboard by providing an indication that they are ready to perform an activity (e.g. human experts raising their hands). A controller then selects one or more of those agents to actually invoke based on some criteria that can be either pre-established or dynamically determined [Nii, 1986]. This opportunistic control can be very effective in domains where there exists a great deal of uncertainty about whether the information currently on the blackboard is correct.

Although opportunistic processing is the traditional blackboard control style, it is also extremely effective at reacting to external and exceptional events. External events occur in domains where the environment can be influenced by forces that are not under control of the blackboard system. Exceptional events can occur in any complex domain where something happens at an unanticipated time or in an unexpected place. For example, an exceptional event in an agricultural application might be the failure of some agent to find a solution under the current problem specification or the violation of a crop residue assumption made by one agent in a solution provided by another agent. Since blackboard agents are triggered directly by events occurring in the environment, external and exceptional events are treated just as any other event and can be dealt with as a natural and inherent part of the system [Lander et al., 1996].

However, opportunistic control can cause several difficulties. Since opportunistic control reevaluates the situation after every operator execution, if there is no representation

of long-term goals or plans, than this can lead to fragmented activity (i.e. it is hard to see the forest for the trees). Furthermore, with opportunistic control, many developers find it difficult to envision and structure appropriately. In agricultural environments, although complex, there are often sequences for problem analyses and solution generation which human specialists follow, that can be applied to accomplish specific goals, and it feels more comfortable to specify those sequences through a workflow, rule-based style notation [Shaffer and Brodahl, 1998]. For many people, a strategy type control style is more natural and easier to visualize than opportunistic control, ensuring coordinated, focused activity among a set of agents.

4.2.6.2 Legacy Data on the Blackboard

Blackboard-based database technology can incorporate legacy data. It is important within the blackboard model to differentiate between legacy software systems and legacy data, as both are present within the blackboard system paradigm. Legacy software systems can be classified as reusable software. Legacy data is data that has been used but persists in the blackboard database to be reused.

The global blackboard database is specifically designed to provide efficient global information management of a shared repository containing various types of global information required in a problem-solving environment. This information could include, problem specifications, partial and completed solutions, suggestions as to how to proceed with a problem, information about the environment itself, and evaluations of completed and in-progress work. Hence, objects (i.e. information and data) in the blackboard database can and do remain as legacy data for long periods of time. If it were not for this legacy data, the blackboard database could be eliminated by introducing direct calls among agents by a configuration-time compiler. The legacy data within blackboard objects is indicative of how the blackboard database serves as a global "memory" for the agents. Objects remain in the blackboard database to be used when and if they are needed by the agents. Without the global blackboard database and legacy data, each agent would be forced to maintain its own copy of objects received from other agents, making the goals of "open system" and "reusable software" impossible.

Legacy data are implemented primarily by the blackboard database operations. The two significant operators for managing legacy data are insertion and merging. When a blackboard object is created, it must be placed in the blackboard database by an insert operation. This is accomplished by creating one or more locators (i.e. pointers) that are used to retrieve the object. Multiple locators are used to support efficient retrieval of objects based on complex criteria. These locators are determined based on attribute values of the object. When an object is placed in the blackboard database, it is important to determine if an identical object already exists there.

4.2.7 Active Databases

A problem which requires cooperation among problem-solving agents, in order to be solved, is sometimes described as an activity. An activity can be segmented either statically or dynamically into subactivities. Whenever an activity is input to the problem-solving system, the activity is decomposed into subactivities each of which is completely within the scope of some problem-solving agent. Consequently, matchmaking is performed between the requirements (i.e. data, knowledge, and solution techniques) of the subactivities and the capabilities of the problem-solving agents. Since the problem-solving agent's capabilities are stored and maintained in a database, the task of selecting the correct problem-solving agent for the subactivity reduces, in simple cases, to querying the capability database (e.g. meta-data). A coordination plan is generated for each activity either as part of the activity specifications or as part of the problem-solving process. A coordination plan specifies and relates the temporal, data, knowledge, and synchronization dependencies between the problem-solving agents chosen for an activity. This coordination plan is created by a translation module which analyzes the activity specification by taking into consideration task dependencies, possible parallelism among subactivities, contingency plans in case of time-outs, failures etc.

In a cooperative problem-solving environment, coordination requires knowledge of the actions of other problem-solving agents, and the ability to reason about the effect of those actions. Hence, coordination is most likely to be derived dynamically, necessitating dynamic coordination. One methodology for dynamic coordination is through the generation of event-condition-action rules from activity specifications. Since coordination

and problem solving are done dynamically, it is not possible to predetermine interactions among problem-solving agents. Consequently, problem-solving agent interaction needs to be based on the sequence of actions, which necessitates that the underlying problem-solving agents need to support various types of events. It is believed that adapting techniques in the area of active databases can provide this capability.

Active databases have been defined as database systems that respond automatically, without user interaction, to events that are generated either internal or external to the system itself [Medeiros and Pfeffer, 1990]. Database system responses are declaratively expressed using event-condition-action rules proposed by Dayal et al. (1986). Event-condition-action rules are comprised of an event that triggers the rule, a condition describing a given situation, and an action to be performed if the condition is satisfied [Diaz et al., 1991]. Therefore, the system not only knows how to perform operations, but also when operations need to be performed.

The structure of the active database is determined by the specification of subactivities and their interactions that are required to support the plan for managing and executing the set of activities in the environment. An active database system services rules that are formed by events, conditions, and actions. Basically, when an event occurs, a condition is evaluated and an appropriate action is taken. The plan defines the set of circumstances that can arise while processing a set of activities and the set of actions or rules that need to be executed as a result of those circumstances. The role of the active database is extremely important to cooperative problem solving, as without it, this functionality needs to be incorporated into the activity coordinator, thereby unnecessarily increasing its complexity.

Research has been conducted on active behavior in the areas of programming languages, artificial intelligence, and database technology. A programming language called ACTOR [Hewitt, 1977] was a pioneer in providing objects with active behavior. In artificial intelligence, active behavior is provided through daemons and active values. Active capabilities have been used in relational databases to enforce integrity constraints, define views, translate update requests and computer derived attributes [Eswaran and Chamberlin, 1975; Stonebraker et al., 1990; Morgenstern, 1984]. Stonebraker et al.

(1991), saw rules as a unifying paradigm for providing a broad range of database facilities. However, in relational databases, rules are implemented as a separate and distinct layer requiring additional mechanisms and structures to support rule management.

4.2.7.1 Object-Oriented Modeling Technology

Object-oriented modeling provides a different approach to traditional system design. Whereas procedural design emphasizes the decomposition of the problem into a set of sequentially executable tasks, the object-oriented design focuses on the entities involved and how they interact with each other [Rumbaugh et al., 1991]. The object-oriented modeling technology is flexible and versatile providing a seamless basis for designing systems and programming code, but can also be used to design databases [Rumbaugh et al., 1991]. Object-oriented database designs are efficient, coherent, and less prone to the update problems experienced by many other database design techniques. In addition, the use of a uniform object-oriented design technique improves integration of database and programming language code. The object-oriented modeling approach allows the same concepts and notation to be used throughout the entire software engineering process.

The fundamental idea of using objects is to raise the level of abstraction, producing databases that comprise encapsulated objects (e.g. weed objects that "understand" population thresholds, density, and herbicide residue) rather than having to establish a framework of relational tables, tuple updates, and foreign keys, etc. to establish the same understanding. However, some developers will argue that even though objects are the natural unit of security and authorization, of recovery, and of concurrency, these advantages are useless in the instances where integrity constraints span objects.

In order to achieve efficient linking of rule management (i.e. essentially a knowledge-base) with an active object-oriented database system, in a problem-solving environment, it is essential that modeling of data and knowledge relationships be properly represented. The design methodologies employed for this type of integration of "bases" are usually based on conceptual data modeling and knowledge representation techniques. Recently, object-oriented modeling techniques have been proposed to represent these knowledge/data relationships and thereby support the design of coupled knowledge-base and database systems. Such systems will take advantage of the natural abstraction

representation, data, and operation encapsulation, and superclass/subclass inheritance features of object-oriented technology.

The literature indicates that previous methodologies have not explicitly addressed the modeling mechanisms required to deduce problem-solving knowledge from object relationships nor, especially, to represent the control knowledge required for object-oriented inferencing that can readily be implemented in object-oriented environments. It is anticipated that rule management can contribute to these problem-solving and control knowledge requirements. The following is a brief examination of the constructs used to represent and support structural knowledge.

Constructs for the structural knowledge can be developed from object-oriented concepts. Objects and classes, consisting of attributes and operations, can represent the structural entity knowledge, whereas the structural relationship knowledge can be represented using the three object-oriented relationships: aggregation, association and generalization. Three constructs are briefly examined below.

1) Constructs for the General Procedural Knowledge

Procedures can be used to represent the general procedural knowledge which refers to data operations performed on classes. Procedures can be encapsulated within classes and hence procedures of a superclass will be inherited by its subclasses. The format for a procedure can be:

Procedure-name (input parameters):

Process: implementation description of the procedure

2) Construct for the Heuristic Knowledge

Rules will be used to represent the heuristic knowledge. Each class will have a set of rules which describe the heuristic pertaining to that class. Like operations of a class, rules can be applied to objects in the class. Through the object-oriented inheritance concept, the rules in a class can be inherited by its subclasses. This should provide a method for structuring rules within class hierarchies. The format for a rule can be:

Rule-name

If: specification of condition

Then: specification of action

3) Construct for the Control Knowledge

Control knowledge is concerned with the process of heuristic knowledge reasoning. Since the heuristic knowledge in an object-oriented approach is structured using class hierarchies, the corresponding control knowledge includes searching within class hierarchies, dynamically creating required sets of objects, processing heuristic knowledge, that is rules, of all created objects and then deleting all these objects. The creation and deletion of these domain-independent objects requires more study. However, the general idea is as follows:

Classes will own triggers, for the purposes of searching within class hierarchies and dynamically creating objects. Searching within one class hierarchy will be enabled by the trigger of a superclass that determines an appropriate subclass through the generalization relationship and activates the appropriate subclass's trigger. This would result in making the heuristic knowledge of a more specialized class accessible. Whether association or aggregation relationships exist between two class hierarchies will govern whether searching from one class hierarchy to another is allowed. This type of searching could be accomplished by having a trigger in one class hierarchy activate a trigger in another class hierarchy. Objects can be dynamically created by triggers during these search processes. Hence, triggers can be treated as a dynamic object creation mechanism. By dynamically creating objects, heuristic knowledge of the required objects can be derived and then be applied. The format for a trigger can be:

Trigger-name (arguments):

When: activated condition

Do: action specification

4.2.7.2 Rule Management

The object-oriented modeling approach of Rumbaugh et al. (1991) allowed the same concepts and notation to be used throughout the entire software engineering process. It would be improper to introduce to the object-oriented modeling concepts an additional mechanism or a different structure to enable support for rules. Hence, rules must be defined and treated like other objects in the system to maintain a unified approach.

Rules are seen as first-class objects, and are defined and described using attributes and methods [Diaz et al., 1991]. All systems do not necessarily consider events as first-class objects. For instance, events can be treated as simple attribute values. However, this approach can compromise the extensibility of the system to cope with events coming from different sources, or events that need special treatment, such as composite events [Dayal, 1989]. Rule management operations and procedures are conceived and implemented as methods. This provides rule management with all the advantages of the object-oriented paradigm. In a unified approach, the system should not distinguish rules from other kinds of objects. Hence, rules can be related to other objects as well as arranged in hierarchies. Methods, which are attached to objects, can trigger rules, and rules are themselves objects. Hence, rules can be defined which are triggered by methods which in turn are attached to rules [Diaz et al., 1991]. As with any entity in the object-oriented paradigm, the meaning of a rule resides in the attributes attached to the rule, and their interpretation by the associated methods. However, from the system's point of view, no distinction should be made. With rules being treated as objects, there is the advantage that any new facility introduced for objects is automatically applicable to rules (e.g. transaction mechanisms, locking mechanisms, display facilities [Diaz et al., 1991]).

There are several object-oriented systems described in the literature that support rules [Kotz et. al., 1988; Dayal, 1989; Hudson and King, 1990; Chakravarthy, 1989; Medeiros and Pfeffer, 1990]. Medeiros and Pfeffer (1990) provided a review of different mechanisms for supporting rules including:

- 1) Method-based mechanisms whereby the rule is precompiled at each location in the code where it might be activated. Alternatively, commands could be placed in the code to fire the rule whenever applicable;

- 2) Object-based mechanisms which enlarge the object description to indicate which rule to involve whenever message sending is initiated;
- 3) External mechanisms which define additional structures which support checking when some event occurs.

Several drawbacks of the first approach can be listed [Diaz et al., 1991].

- a) With the rules buried inside methods, it is difficult to inquire about any of the rule's attributes such as the condition, the action, or whether it is enabled or not.
- b) Modification to any of the attributes of a rule requires making changes in every method supporting the rule.
- c) Since rules can interact, coding of rules within methods requires the programmer to understand all the rules that appear in the method to ensure that interaction can be handled properly.
- d) The rule definition is dissipated in different places compromising the object-oriented philosophy that encourages all information about a given object to be gathered in one place.
- e) Method coding now includes two aspects, how the operation itself is implemented and the enforcement of the rule. This severely compromises method overriding which is a useful mechanism in object-oriented systems for customizing an operational implementation for special requirements. The problem is that in this case not only is the operation being overridden but also the embedded concept described by the rule (e.g. possibly an integrity constraint).

In Stonebraker et al. (1991) some of these points are addressed and it was indicated that only one reasonable solution exists and that is that rules must be enforced by the database management system, but not bound to any function (i.e. method) or collection. The remaining two approaches for supporting rules (listed on the previous page) overcome these disadvantages by providing a mechanism that is supported by the database management system.

Since any message can raise an event, evaluation of rules imposes an overhead on every possible event that can be detected by the system. In relational databases, events are generally restricted to be database updates. Hence, the need for efficiency for rule support in object-oriented databases is greater than with relational databases. However, Diaz et al. (1991) attempted to enhance system performance by indexing rules by class. In so doing, the search for applicable rules is considerably reduced increasing system performance.

In Medeiros and Pfeffer (1990) a rule management approach was proposed for a system called O2. With this approach, rules are objects having the event as an attribute and auxiliary structures are defined for storing rule lists which are checked when specific events occur. However, events are not viewed as objects in themselves, and thus, system extensibility can be compromised in the sense that composite events or events with special requirements are difficult to introduce. Furthermore, a local mechanism is used to provide rule "inheritance" instead of using a mechanism, such as the generalization relationship [Rumbaugh et al., 1991], which is based on the object hierarchy itself.

Dayal et al. (1986), in their system HiPAC, envision rules and events as different entities with their own attributes and methods. Within HiPAC, a sound approach is taken in support of rules, paying special attention to transaction management and optimization techniques. However, with HiPAC, some of the idiosyncrasies of the object-oriented paradigm have not been considered, such as the primary role that classes play, where methods are part of the class definition.

In order to provide a rule manager in the context of an object-oriented database system, a primary requirement is to identify the significant entities and their interaction. The function of a rule manager is to provide quick response to events which have been generated by some system, by utilizing rules. Three components can be identified in this process [Diaz et al., 1991].

- 1) A rule describes both when and how the system is to react to an event.
- 2) An event is a flag to signal that a specific situation has been reached to which reactions may be necessary [Kotz, 1988]. All systems do not consider events as first-class objects. For instance, events can be treated as simple attribute values. However, this approach can compromise the extensibility of the

system to cope with events coming from different sources, or events that need special treatment, such as composite events [Dayal, 1989].

- 3) The event generator can be viewed as any system producing events which may need a special response in terms of rule triggering. Events can be generated by the database management system itself or by some other external system such as a clock or an application program.

The relationship between these three entities can be described as follows [Diaz et al., 1991]. First, a rule can be triggered by an event, however, an event can trigger several rules. Second, an event can be generated by several systems, and a system can generate several events. The major interaction between these entities can be described as follows:

- 1) An event is created by any event-generator and is indicated to the event manager through the message signal.
- 2) The event manager checks if any rule can be triggered by the event signaled, and if so, it sends the message "fire" to the appropriate rules.
- 3) When the message "fire" is received by a rule, the rule condition is then checked and if satisfied the rule action is executed.

Other types of interactions are also possible, such as "awakening" of events as a result of rule creation [Diaz et al., 1991].

In relational databases, an event can be described by the operation together with the time when this operation takes place, that is, before or after [Davis, 1993]. For example, the pair (*insert, before*) could specify that the event arises *before* the operation *insert* occurs. In this context, object-oriented databases have some differences from relational databases. In object-oriented database systems, operations (i.e. methods) are not isolated but are part of the class definition [Chakravarthy et al., 1993]. The class is not just an argument of the method, but rather the method itself is subordinated to the class. This can lead to several results; the same method name can be implemented in different ways in distinct classes; the process known as overloading can occur; or a method can be inherited down the hierarchy by any subclasses, thereby revising the behavior of the superclass.

A problem with incorporating rule management within active databases requires that there be a specific protocol for communications among the agents. Depending on the

complexity of the problem being solved, this may not be a major problem if an accurate coordination plan can be developed in a sequential manner. However, if the problem solving is complex where coordination requires knowledge of the actions of other agents and the ability to reason about the effect of those actions, then coordination must be derived dynamically necessitating that a dynamic coordination methodology be developed.

4.3 Conclusion

The literature review and research clearly indicate that methods, technologies, and theories exist from which a computer-based conservation oriented crop production decision support system could be developed by integrating individual expert systems such that they cooperate in a problem-solving environment to produce results and recommendations to aid the client's (e.g. producer's, extension specialist's) decision-making process. Such a decision support system could provide the required technology transfer and aid the technology adoption process as discussed in the previous sections. The remainder of this dissertation describes a conservation farming decision support system called Farm Smart 2000. The three main goals and objectives of the development of Farm Smart 2000 were as follows:

- 1) **Provide Cooperative Multi-Agent Problem Solving** - The complexity of problems that can be solved by cooperative multi-agent problem-solving systems far exceeds the problem-solving capabilities that could be obtained by developing one large computer-based knowledge system. A specialized multi-agent system which shares expertise has several advantages over a large monolithic system developed to solve the same problem(s).
- 2) **Open System Architecture** - The development of an open system to facilitate ongoing technology transfer. Farm Smart 2000 is a vehicle or mechanism for transferring technology and knowledge from the research laboratories and field plots to the end-user. In so doing, Farm Smart 2000 must support independent expert system development .
- 3) **Provide Decision Support Specific to Individual Farming Systems** - This goal is essential to the overall success of Farm Smart 2000. Without it, Farm

Smart 2000 has minimum usefulness and effect on the producer's decision making, resulting in minimal credibility.

Chapter 5: The Development of Farm Smart 2000

Farm Smart 2000 consists of three tiers of decision support with tier-3 being the focus of this dissertation. One of the factors for success of a decision support system is to establish and maintain credibility with the end-users. Hence, it is important that Farm Smart 2000 be able to provide decision support that is specific to farming systems of the individual producers. Farm Smart 2000 is built on an open system architecture utilizing multiple agents in a cooperative problem-solving environment. A Global Control Expert and a blackboard model help with these achievements.

5.1 Overview of Farm Smart 2000's Three Tiers of Decision Support

Farm Smart 2000 as illustrated in Figure 5.1 provides accurate and meaningful decision support through three tiers, from the "informational" level (i.e. tier-1) to the "expert" level (tier-3) with tiers 2 and 3 providing specific decision support to the farm manager's operation, given that sufficient on-farm data is available. If sufficient information is provided to a human extension specialist, he can provide specific, accurate and unbiased decision support because there is no need to make assumptions or interpretations. Similarly, the more on-farm data and information available to Farm Smart 2000's tier-3, the "smarter" it becomes, providing specific, accurate and unbiased decision support.

At the low level of decision support (i.e. tier-1), called Basic Decision Support, the growing acceptance of Internet web sites provides Farm Smart 2000 with the capability to disseminate timely information creating a delivery mechanism for general or "basic" decision support. Farm Smart 2000's web site organizes and provides a wealth of agronomic information based on various items including factsheets, newsletters, informational bulletins, answers to frequently asked questions, and a specialist directory called Who's Who. Tier-2, called Advanced Decision Support, is a collection of stand-alone expert systems, each focused at providing decision support for specific narrowly defined problems.

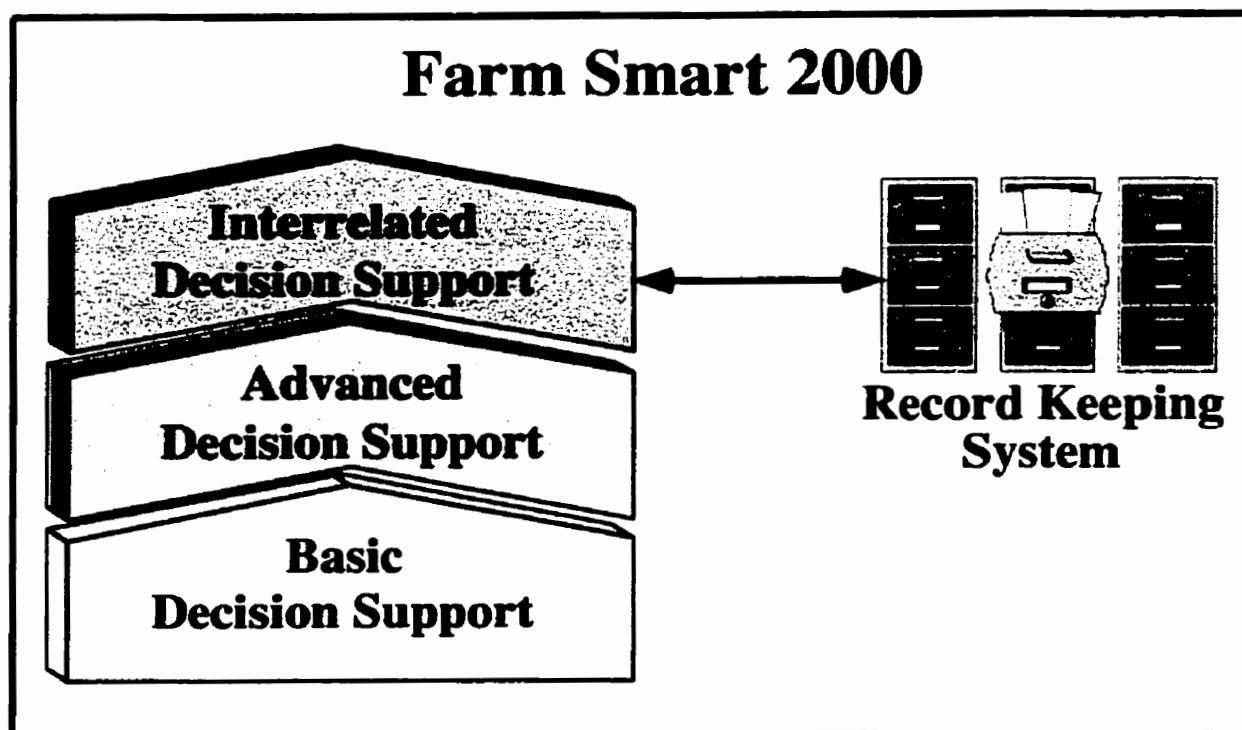


Figure 5.1: Farm Smart 2000's Three Tiers of Decision Support

At the top level (i.e. tier-3), called Interrelated Decision Support, Farm Smart 2000 integrates multiple knowledge sources (e.g. expert systems and knowledge-bases) into one comprehensive decision support component. Not only can Farm Smart 2000's tier-3 provide pertinent farm-specific decision-making support, but also detailed explanations of reasoning procedures to rationalize the recommendations specified. The Interrelated Decision Support is called Crop Planning in the actual Farm Smart 2000 system.

A suitable existing record keeping system that would fulfill the requirements of Farm Smart 2000 was sought for reuse, so as not to "reinvent the wheel". However, such a record keeping system was not available and consequently a customized one called SmartRKS was developed specifically for use with Farm Smart 2000. Farm Smart 2000 through its 3-tier approach with its record keeping system, as illustrated in Figure 5.1, provides a variety of agricultural decision support information and knowledge obtained from numerous sources in various formats. Furthermore, with its 3-tier infrastructure, there is no limit to the type of information or knowledge that can be utilized, nor any need to screen information and knowledge for its ease of integration, but rather, utilize it in

which ever tier it fits best. The focus of this dissertation is tier-3, with tier-1 and tier-2 being discussed in the following paragraphs only to the extent required to aid the reader's understanding of the more detailed description of Interrelated Decision Support (i.e. tier-3).

5.1.1 Basic Decision Support

In tier-1, the Basic Decision Support level, information is provided through a web site (<http://paridss.usask.ca>). As with all successful web sites, constant maintenance and updating are required for Farm Smart 2000's web site to remain useful. To remain current, Farm Smart 2000's web site implements an on-line automatic "self-update" methodology to permit authorized users to create and update their own information using individual password security. This reduces the burden on the web site's administrator, while ensuring that information is up-to-date, and thereby encouraging visitors to visit the site often where timely options for Prairie farmers are provided to create economic opportunities. For example, through specific "administration" web pages of the Farm Smart 2000 site, authorized users (i.e. those with a username and password) can log on and perform updates to web pages to which their passwords provide access.

For instance, several information specialists of AAFC have authorization to access and update Events, What's New, and Related Links. When they access the administration pages they are presented with a list of choices to update with the capability to add, delete or replace. They enter the description of the event, for instance, in a text box, the date when the event begins and ends, contact information, and click the submit button. The new event is added to the appropriate page and after the event is over, this event entry will automatically be deleted by the system based on the "end date".

In addition, Farm Smart 2000's web site has an advantage over other similar sites in that Farm Smart 2000 collaborates with producer organizations and associations, providing them services necessary for the storage and public access of their Internet web pages. In return, the Farm Smart 2000 web site receives the opportunity to host these organization's web pages, thereby enhancing and expanding tier-1's decision support information-base and also providing a central Prairie web site where information from all the major soil conservation organizations is available.

Farm Smart 2000's Internet home page incorporates three main areas: partners, aspects directly related to Farm Smart 2000, and a quick-go-to menu of general frequently accessed information. The partners currently comprise the following:

- Alberta Reduced Tillage Initiative (ARTI) in Edmonton, Alberta;
- Conservation Learning Centre (CLC) south of Prince Albert, Saskatchewan;
- Indian Head Agricultural Research Foundation (IHARF) at Indian Head, Saskatchewan;
- Indian Head Precision Farm Centre at Indian Head, Saskatchewan;
- Prairie Crop Protection Planner at the U of S, Extension Division, Saskatoon, Saskatchewan;
- SSCA at Indian Head, Saskatchewan;
- Seager Wheeler Farm at Rosthern, Saskatchewan;
- Specialized Crop Production at Indian Head, Saskatchewan.

The second main area pertains to aspects directly related to Farm Smart 2000 and includes the following (* designates items having on-line automatic "self-update" functionality):

- What's New* - pages to post new information including important release dates, new software, enhancements and similar announcements;
- Who's Who* - is a directory of experts which can be searched and contains information including contact information, area(s) of expertise and areas of general knowledge and interests;
- Research Datasets* - is a data dictionary providing the description and documentation of research databases, datasets, models and reports which are available either on-line or directly from the researcher, in the case of maintaining intellectual property rights;
- Related Links* - are common on Internet sites and provide the user with links to other sites where related information can be obtained;
- Acknowledgments - acknowledges others for their participation and contributions to Farm Smart 2000's web site;

- Feedback - provides the functionality for users to comment on the web site or make inquiries.

General information that is accessed frequently is readily available on the Farm Smart 2000 home page and includes the following (* designates items having on-line automatic “self-update” functionality):

- Events* - defaults to Farm Smart 2000’s events, but all the partners’ events are also available;
- Frequently Asked Questions (FAQ)* - defaults to Farm Smart 2000’s FAQ’s, but partners’ with FAQ sections are also available;
- Newsletters* - defaults to Farm Smart 2000’s newsletters with partners’ newsletters also available;
- Fact-Sheets* - defaults to Farm Smart 2000’s fact-sheets with fact-sheets of partners also being available;
- Reports* - defaults to Farm Smart 2000’s reports but also has the reports of all the partners available;
- Search - provides a search engine enabling the entire Farm Smart 2000 site to be searched.

5.1.2 Advanced Decision Support

In the Advanced Decision Support level, individual stand-alone expert systems contained in a software “tool box”, quickly and easily aid farmers in solving narrowly defined problems (e.g. Which variety should I grow with my soil conditions? or How do I control this pest?). The user’s “tool box”, to which the user can add and remove “tools” (e.g. expert systems or any other application software) is distributed on CD-ROM and can be configured with any of the following available Farm Smart 2000 expert systems:

- AFFIRM - The Alberta Farm Fertilizer Information and Recommendation Manager;
- ASK - An Agronomic Soil Conservation knowledge-base;
- ASSESS - A Soil Salinity Expert System;
- Crop Rotation Planner - An expert system for crop rotation planning;
- Crop Protection Planner - An expert system for crop protection planning;

- Crop Variety Select - An expert system for selecting Canadian prairie crop varieties;
- CCS - A Climate Control expert System;
- DISTA - A plant disease expert system;
- PARMS and RTDS - Planting and Residue Management expert systems;
- SoilCrop - An expert system for soil conservation crop productivity relationships;
- STARRT - Stepwise Technology Adoption Risk Reduction Tool;
- WMP - A Weed Management Planner expert system consisting of 6 modules including a Problem Weed module, Long Term Management module, Weed Identification, Weed Survey module, an Economic Threshold module, Weed Density Map module, and a link to the Prairie Crop Protection Planner.

In addition, the "tool box" provides a "storehouse" for expert systems where they can be utilized and tested before being integrated into tier-3.

5.1.3 Interrelated Decision Support

In tier-3, the Interrelated Decision Support, broadly defined problems are supported through a fully integrated system in which farm management knowledge is interrelated with agronomic information to provide decision support. For example, tier-3 supports decisions for planning which crops to grow on what fields during a growing season. This interrelated approach is unique among current agricultural decision support systems. To accomplish this, Farm Smart 2000's tier-3 integrates multiple expert systems/agents in such a manner that they cooperate to solve complex interrelated crop production problems. This cooperation is achieved through an open architecture allowing individual agents to interact with a Global Control Expert. The Global Control Expert manages the transfer of data between agents and knowledge-bases, and allows for the integration of new knowledge sources and agents to enhance problem solving of complex issues. Hence, Farm Smart

2000 is open to new knowledge and research results and can transfer such technology effectively to the producer without system re-engineering.

Although day-to-day or week-to-week problems are usually best supported by tiers 1 and 2, the greatest strength of the Interrelated Decision Support is planning and problem-solving over the long-term. For instance, various planning scenarios for a growing season can be analyzed and stored, and later retrieved for comparison or for more "if-then" analyses and planning. To aid planning and problem-solving in the Interrelated Decision Support tier, goals and constraints can be set by the user through pick lists which enables the system to focus on the user's specific situations and biases. Setting goals and constraints allows the system to narrow the available solution options, resulting in more targeted solutions.

A feature of Farm Smart 2000's problem solving is its capability to detect other relevant concerns and "flag" them for the user. For instance, Farm Smart 2000 might recommend applying a particular pesticide to a field that it knows (from the producer's record keeping system) is adjacent to a pasture. In this case, the system would flag the concern of this pesticide being harmful to cattle if they are grazing in the adjacent pasture when the pesticide is applied.

Tier-3 provides Interrelated Decision Support over a broad spectrum of concerns through the integration of multiple reusable agents utilizing the architecture introduced earlier called a "blackboard" to provide a cooperative problem-solving environment. Using this same methodology from the blackboard analogy, but with artificial intelligence, tier-3 is able to solve problems which span several knowledge areas by utilizing knowledge-bases, rule-sets, and on-farm data, through friendly interaction with the end-user. There are four main agents and knowledge sources currently integrated in tier-3 including the Weed Management Planner, Crop Protection Planner, Crop Variety Select, and Crop Rotation as illustrated in Figure 5.2.

Farm Smart 2000, with its 3-tier infrastructure, provides the capabilities and flexibility necessary to solve problems of different magnitudes, from narrowly focused specific problems to broader planning problems. Although the individual expert systems in

tier-2 are efficient at solving specific problems, the Interrelated Decision Support tier can also solve specific problems by utilizing only the data areas (i.e. folders) necessary.

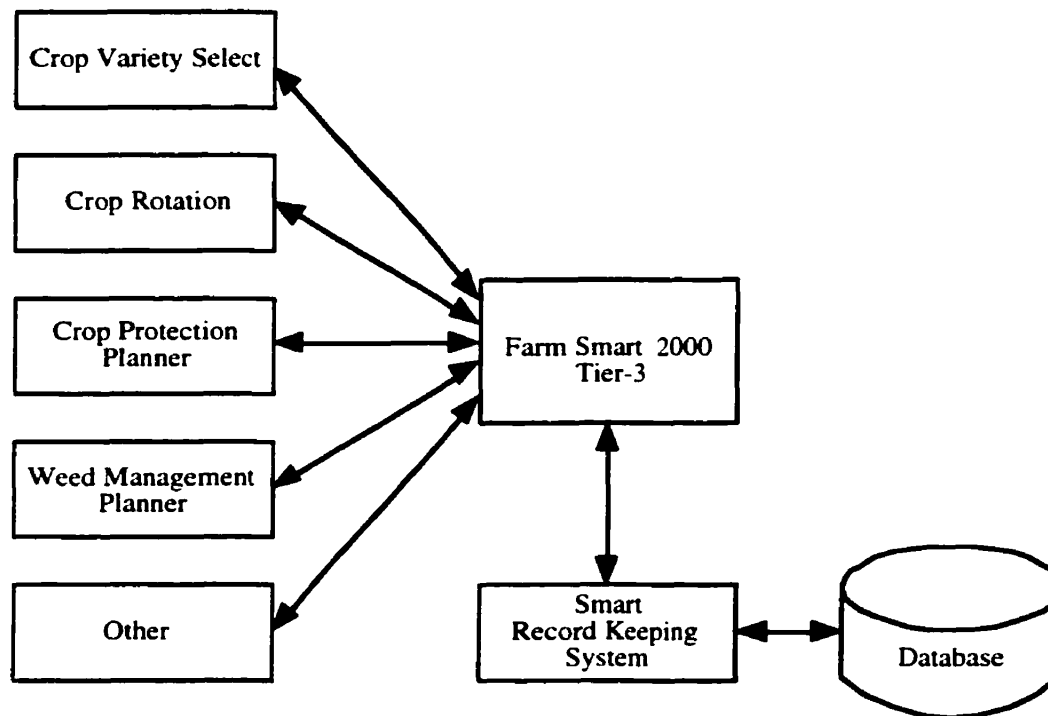


Figure 5.2: Farm Smart 2000 Tier-3 System Architecture

5.2 Development of Farm Smart 2000

The entire development of Farm Smart 2000 was a team effort engaging experts from provincial and federal governments, universities, agricultural associations and societies, private industry and crop production enterprises (i.e. farmers). These team members participated at various levels of development including the “needs” assessment, specifications feedback, on-going consultation, development of expert systems, and testing and evaluation of software.

Three main goals were established for the development of Farm Smart 2000 as stated in Section 4.3. As indicated in Section 5.1, the main focus of this research is tier-3, the Interrelated Decision Support level as illustrated in Figure 5.3 (in the area within dashed lines). However, some of the goals overlap each other, and multiple tiers. The emphasis of the development of Interrelated Decision Support is to research the integration of heterogeneous reusable agents in the cooperative problem-solving paradigm utilizing global system representation and object-oriented database techniques, supplemented with

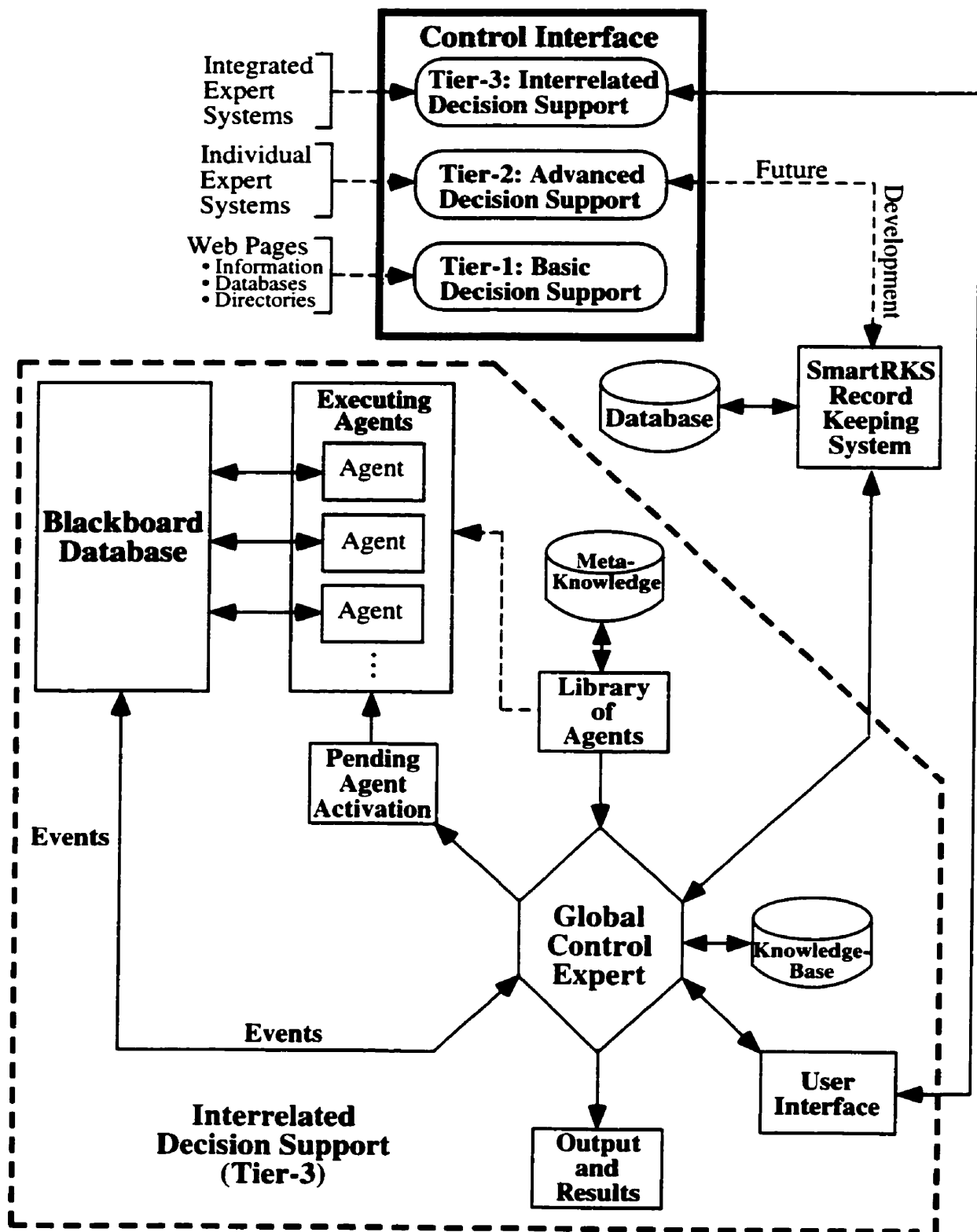


Figure 5.3: Farm Smart 2000 Components

enhanced blackboard-based database and rule management methodology to achieve an open architecture. The following sections will discuss the three main goals (See Section 4.3) and the method(s) of solutions used.

5.2.1 Providing Cooperative Multi-Agent Problem Solving

The requirements for cooperative multi-agent integration in tier-3 (i.e. goal #1, Section 4.3) were initially analyzed and a high level (i.e. more abstract, less user specific) methodology was developed. A corresponding prototype application was implemented (i.e. PARI DSS '95) using a top-down approach to integrating multiple heterogeneous reusable agents based on conflict resolution techniques as the basis for system coordination. In this prototype, the user selected from a list, crop production questions which best matched their particular problem and then the system solved the problem (i.e. answered the question) based on user input. This initial prototype application was too limiting, in that it could not adequately represent the complex interrelations of conservation farming systems resulting in providing only general decision support for specific problems. A bottom-up approach utilizing data and knowledge-bases was researched which is the methodology that ultimately was used in Farm Smart 2000. Tier-3 was redesigned using a bottom-up approach where underlying on-farm data and agronomic knowledge are the basis behind problem and planning solutions. A bottom-up approach is developed by determining solutions to smaller simpler problems (i.e. sub-problems) and then extending these existing solutions to more complex problems in an incremental manner within the cooperative problem-solving paradigm.

Farm Smart 2000 required the development of a cooperative multi-agent problem-solving system. Developing all agents with inherent knowledge and information about all other agents such that agents interact directly with each other through a collective action and take action or make decisions based on influence from or knowledge of other agents, is not possible in Farm Smart 2000 as it negates several of the goals previously stated. This approach would prohibit the development of an open system and individual independent expert systems. Rather, a global structure is required containing information specifically about each agent (i.e. meta-knowledge) to use for agent coordination and communication during the problem-solving process and providing global control, global consistent

knowledge, and globally shared goals. Hence, in Farm Smart 2000, the agents are heterogeneous and reusable with separate individual goals, and are integrated through a cooperative problem-solving environment to provide interrelated comprehensive decision support.

In the paradigm of cooperating expert systems, heterogeneous reusable agents must be integrated such that the state of problem-solving is communicated and agents' actions are coordinated to arrive at mutually acceptable solutions. In these cooperative environments, conflicts must be resolved as a result of incomplete or inconsistent knowledge and/or incorrect assumptions, different problem-solving techniques, and different solution evaluation criteria. Anticipating and removing potential conflicts through software engineering at agent-development time is not always possible since it is not known what knowledge will be contained in an open system. Hence, conflicts must sometimes be detected at time of program execution and resolved dynamically. Software reuse in any form is hindered by the absence of technical tools and techniques to support information sharing [Neches et al., 1991]. Conflict resolution is accomplished in Farm Smart 2000 through the selection and development of agronomic goal and constraint rule-sets which are designed to be activated utilizing active database technology. Through the generation of activity rules within the global system representation, dynamic coordination of activities can be achieved.

Tier-3 of Farm Smart 2000 utilizes reusable expert systems, and with minimal modification to them, integrates them such that they cooperate to solve complex agricultural problems. This is not an easy task as stated by Lander and Lesser (1994, p. 13); "There is a high degree of complexity inherent in building heterogeneous agents that can understand each other well enough to positively affect mutual work." There are tradeoffs between the two approaches described above for developing cooperative multi-agent problem-solving systems. With the global structure approach, information must be specifically defined for each agent and corresponding coordination strategies, albeit, all this information is contained in one place. However, with the contrasting approach, each agent must be developed specifically with its own set of information and knowledge about each other

agent, including translation capabilities, in order for the agents to interact directly with each other.

5.2.1.1 Managing Agents

The approach whereby each agent has a built-in set of information and knowledge about each other agent, including translation capabilities, defies the term reusable agent, and hence, was not acceptable in the development of tier-3. The global structure approach held more promise for achieving the required communication and coordination among the agents such that the open-system goal could also be met. Adding a new agent only requires building an "agent information file" with which to inform the global control mechanism (i.e. the moderator in the human analogy) what problems this new agent is capable of solving.

The "agent information file" is actually meta-knowledge contained in a Global Control Knowledge-base. As Farm Smart 2000 changes, with the addition of new agents and deletion of agents no longer required, the meta-knowledge within the Global Control Knowledge-base needs to be updated. The objective of the Global Control Expert, when using meta-knowledge, is not to locate a specific solution, but to focus itself in its search for a solution based on the expertise held by the agents available in the system [Lander and Lesser, 1994]. In order for the Global Control Expert to guide the processing towards an acceptable result, it must be able to determine the agents' requirements for solutions. This is accomplished by utilizing the agents' meta-knowledge. Agent meta-knowledge consists of knowledge including:

- what are the agents' goals;
- what implicit heuristics are employed;
- what results is the agent capable of providing;
- what are the agents' input parameters;
- what are the agents' conflict thresholds;
- what are the agent's weighted constraints;
- how are constraints imposed.

One of the key entities of the meta-knowledge is constraints. The Global Control Expert must be able to assimilate constraining information of each agent in order to guide the problem-solving activities.

5.2.1.2 Distributed Versus Multi-Agent Systems

Distributed database technology has provided techniques for processing queries and transactions over a distributed environment and guarantee certain properties such as atomicity [Date, 1995]. Distributed cooperative problem solving and multi-agent problem solving concentrate on the techniques and strategies for accomplishing cooperation and problem solving using the notion of agents with or without centralized control, and focuses on collaborative aspects of problem solving [Bond and Gasser, 1988]. Distributed problem solving considers how the work of solving a particular problem can be divided among problem-solving agents that cooperate in dividing and sharing knowledge about the problem and in turn develop a solution for the problem. With multi-agent systems, a collection of intelligent agents work towards coordinating and sharing their goals, skills, and plans to cooperatively solve a problem. Multi-agent systems not only share knowledge about the problem, but must also reason about the type of coordination required among the agents in order to solve the problem. Multi-agent systems are most successful when the problem-solving agents are cooperating in a positive manner and friendly environment rather than tending to work against each other, or not being trustworthy, as in a hostile environment, where individual agent goals predominate.

It was evident that the collaboration and coordination offered by distributed database systems is inadequate for employing in Farm Smart 2000. In addition, because of the complicated interrelationships that exist among farming entities, it is impossible for a Farm Smart 2000 problem to be decomposed, with individual sub-problems being distributed to various agents for analysis.

5.2.1.3 Achieving Processing Control

The decision support provided by Farm Smart 2000 is very diverse and consequently an opportunistic control method would lead to excessive fragmented activity resulting in non-conclusive recommendations. Hence, a comprehensive agronomic knowledge-base was developed, coupled with an extensive database containing "farm

specific" information facts and operational data (i.e. a record keeping system), to provide external goal-directed control rather than opportunistic control. Database and knowledge-base technologies are often complementary rather than contradictory. The database maintains well structured data representing the farm and field facts that need to be shared by the different agents, while the agronomic knowledge-base contains less precise and more abstract knowledge used for reasoning about the solution strategy and thereby controlling (dynamically, if need be) the coordination of the cooperative problem solving via the Global Control Expert.

Both goal-directed and data-directed control factors are incorporated in Farm Smart 2000 to provide the effective control required for the coordination of the problem-solving agents. Goal-directed factors inform a system what it would most like to do in order to solve the problem. Data-directed factors inform a system what it is best able to do given the available data. Without goal-directed control in Farm Smart 2000, effort would be wasted pursuing goals that cannot be easily satisfied or pursuing ineffective methods for satisfying goals. Without data-directed control in Farm Smart 2000, effort would be wasted working on data and hypotheses that are not important for meeting system goals.

In addition, goal-directed control simplifies termination criteria, that is, the criteria that must be met for problem solving to terminate. With opportunistic control, determining whether the criteria has been met is referred to as the termination problem, because when termination is viewed as a constraint satisfaction problem, the application is under constrained. For example, there can be multiple potential solutions that are consistent with the constraints. This is often a result of uncertainty in the data and/or the problem-solving knowledge associated with opportunistic control.

5.2.1.4 The Global Control Expert

The focus in this research is problems for which the coordination and collaboration components can be specified. Durfee et al. (1987b) define two types of distributed problem-solving systems. Those that are completely accurate nearly autonomous systems and those that are functionally accurate cooperative systems. The approach taken in this research is applicable for either case, whether the problem-solving agents are autonomous or cooperative. However, if there is cooperation among the problem-solving agents, as in

Farm Smart 2000, then their coordination plan is completely accurate and well defined. That is, there is a specific protocol for communication among agents. Farm Smart 2000, through a Global Control Expert, utilizes predetermined specific problem-solving coordination and communication protocols among agents with a further capability to dynamically alter the coordination if required.

The goal-directed control mechanism in Farm Smart 2000 is structured within a Global Control Expert providing the required coordination. If we view Farm Smart 2000 as a problem-solving team of human experts, a team-leader or manager, having the meta-knowledge of the problem-solving abilities of each human expert, would use his/her knowledge and reasoning abilities to dynamically determine the problem-solving path. Hence, in Farm Smart 2000, the Global Control Expert is the team-leader, providing a mechanism to organize agent use in the most effective and coherent method.

The Global Control Expert provides the global control necessary for interaction, communication and coordination among agents. In addition, the Global Control Expert has access to meta-knowledge about the reusable agents, which is contained within the Global Control Knowledge-base, and used to handle inconsistency and unique beliefs among the agents. In this way, the Global Control Expert is able to reason about the coordinating processes used among the agents in the cooperative environment. The Global Control Expert, by utilizing the Global Control Knowledge-base, allocates tasks to the agents by matching the task requirements with the agent's knowledge and resources. The Global Control Expert triggers agents into action based on influence from the agents' contributions, the knowledge of other agents, and the problem-solving process. Since there is no common communication language shared among the agents, the Global Control Expert monitors the agents' processing of tasks, utilizing the Global Control Knowledge-base, providing the necessary interaction and communication among the agents via the blackboard and a translation data structure. Reasoning on the part of the Global Control Expert is aided by meta-knowledge enabling the Global Control Expert to determine future actions of an agent.

As the agents interact through the Global Control Expert, they post contributions to the shared blackboard. The Global Control Expert detects conflicts when agents post

contributions which are judged to be in disagreement with each other based on the constraints and goals of the original task. In resolving conflict among heterogeneous reusable agents, the Global Control Expert has sufficient artificial intelligence, and access to required knowledge, so as to have a global view of the problem-solving activity with the power to make global decisions that are coherent and in the best interest of the agents involved.

5.2.1.5 Object-Oriented Technology

As discussed in Section 4.2.7.1, object-oriented modeling provides a different approach to system design. Whereas procedural design emphasizes the decomposition of the problem into a set of sequentially executable tasks, the object-oriented design focuses on the different entities in the problem and how they interact with each other [Rumbaugh et al., 1991] which conforms to the objectives of Farm Smart 2000. The object-oriented modeling approach allows the same concepts and notation to be used throughout the entire software engineering process whether it be for designing systems and programming code, or to design databases. Hence, not only are object-oriented database designs efficient, coherent, and less prone to the update problems experienced by other database design techniques, but this uniform design technique improves integration of database and programming language code. With these inherent powerful characteristics and capabilities, it would be improper to introduce to object-oriented modeling, concepts and additional mechanisms, or a different structure, in order to support the use of rules. A rule describes both when and how the system is to react to an event. Hence, rules must be defined and treated like other objects in the system, and then managed, to maintain a unified approach.

In Farm Smart 2000, specific protocols required for agent coordination are predetermined (for the most part, except for conflict resolution) at a higher level of abstraction through techniques including goal, constraint, and priority setting. The approach of utilizing object-oriented databases to assist cooperative problem solving was an alternative. The problem of cooperative problem solving was analyzed from a database perspective and results indicated that enhanced database technology, such as special object-oriented blackboard databases and active database management systems could provide the necessary techniques and abstractions for formulating a viable solution to aid the

cooperative problem-solving approach. The necessity for effective collaboration and coordination was emphasized.

A bottom-up approach is developed by utilizing existing solutions to simpler problems and extending them to more complex problems in an incremental manner. The emphasis with this approach is to utilize artificial intelligent agents and integrate them with traditional database techniques, supplemented with extensible methodology. Object-oriented database techniques and modeling were analyzed and again determined to be the best suited alternatives to supporting a bottom-up approach. Object-oriented technology provides the functionality enhancements required to extend a database management system to support new classes and rule management, both of which were necessary in the cooperative problem-solving environment.

5.2.2 The Development of an Open System Architecture

The development of an open system architecture (i.e. goal #2, Section 4.3) is intended to facilitate ongoing technology transfer. Farm Smart 2000 is a vehicle or mechanism with which to transfer technology and knowledge from the research laboratories and field plots to the end-user (i.e. producer). Farm Smart 2000 turns research data and results into useful information and knowledge, and delivers it to clients. The open architecture multi-agent design of tier-3 makes it "open" to other issues by adding, deleting and/or updating expert systems (i.e. changing the problem-solving environment) without modifying the entire system, thereby enhancing the decision-support capabilities on an ongoing basis. For example, during the course of Farm Smart 2000's development, two different versions of the Prairie Crop Protection Planner, a third party expert system, were integrated into tier-3. Furthermore, support for additional crops and newly developed crop varieties can be added without affecting the entire system.

Although Farm Smart 2000 has guidelines for developing expert systems, there are no stringent standards that must be followed when developing an expert system for use in Farm Smart 2000. Collaborators have a degree of freedom to develop expert systems that fulfill their own needs first which then can be placed into Farm Smart 2000's "toolbox" (i.e. tier-2) and more importantly, integrated into the Interrelated Decision Support Level, with only minor modifications, thus accomplishing a sub-goal to reuse software.

In general, present-day decision support systems are large and complex, reflecting rapid and constant changes to objectives and goals. Often these systems require the expertise of multiple expert systems cooperating to produce comprehensive solutions to complicated problems. These expert systems might be newly developed for each application system, a costly and timely process, or more recently, decision support systems are being built with reusable software (i.e. expert systems and agents). Although this sub-goal, to reuse expert systems/agents, was challenging to accomplish, new technology tools have been introduced which are outlined in Sections 8.4 and 8.5 and need further research to determine if this sub-goal can be accomplished more effectively.

The heterogeneity of agents in Farm Smart 2000 is characterized as both knowledge and implementational. Each reusable agent in the Farm Smart 2000 library of agents uses its own knowledge, goals, priorities, and problem-solving reasoning in determining solutions. Furthermore, these agents were developed independent of each other using different architectures, algorithms, and in some cases different languages.

One strength of Farm Smart 2000 is diversity with its many and varied knowledge sources, not just one, providing a holistic system. The multi-agent system methodology provides Farm Smart 2000 with a natural, flexible, open development framework in which to integrate heterogeneous reusable agents, as well as the capability to implement agents independently with minimal time and resource requirements. Such flexibility brings balance and robustness to Farm Smart 2000, similar to what human experts bring to a project team, providing creative and innovative solutions that are not usually forthcoming from a single expert. Agents developed for multiple uses are generally more reliable and their development cost can be amortized over many uses [Lander and Lesser, 1994]. This approach is encouraging government and businesses, who are being forced and pressured to be more productive with fewer resources, to maximize their large investments in legacy software systems by integrating these systems to improve productivity [Lander, 1995; Lander et al., 1996]. Cost savings can be realized through the implementation of new integrated applications by developing reusable software agents with development and maintenance costs spread over a number of applications.

From a software development perspective, advantages to using heterogeneous reusable agents, which also fulfilled the objectives of this research include [Lander, 1994]:

- 1) the uniformity of expertise within each agent makes them easier to design, develop and maintain than single large systems. For example, if two agents that supply the same output are interchanged, the modification is completely transparent to agents using that output, resulting in nothing being required to propagate the replacement through the environment;
- 2) consistency of knowledge can be maintained within local boundaries without necessitating agreement across boundaries;
- 3) software modifications can be focused at the problem or enhancement within the agent, without requiring changes to be disseminated throughout the entire system.

There are many on-farm problems of different magnitudes requiring decision support. From day-to-day or week-to-week type problems, to problems encompassing one or more growing seasons. Farm Smart 2000's architecture provides the capabilities and flexibility necessary to solve problems of different magnitudes, from narrowly focused specific problems, to broader planning problems. The 3-tier structure of Farm Smart 2000's open architecture does not limit the type of information or knowledge that can be added and utilized. Depending on its characteristics, information and knowledge can be incorporated as basic, advanced or as interrelated decision support.

5.2.2.1 The Blackboard Model's Contributions

Although the blackboard model introduced in Section 4.2.5 is described as a tool for developing cooperative problem-solving systems, it was adopted in the development of Farm Smart 2000 as a framework for developing an open system architecture [Corkill et al., 1988]. Nii (1986) notes that the blackboard model incorporating knowledge sources, global blackboard databases, and control components, does not specify a methodology for developing a blackboard system for a particular application.

The development of the blackboard model was motivated by the need for flexibility in reasoning and for information sharing. The blackboard model is useful in a broad range of applications and is able to facilitate different problem-solving methods. In particular, it

was used within Farm Smart 2000 because of its capability to handle the following problem characteristics [Jagannathan et al., 1989].

With regard to developing cooperative problem-solving systems, the blackboard model can:

- handle a huge solution space;
- fulfill the need for cooperation and coordination among heterogeneous reusable agents in forming credible solutions or recommendations.

With regard to developing an open system architecture, the blackboard model can:

- handle various input data;
- accomplish the integration of diverse information;
- fulfill the requirement for an evolutionary solution.

In a blackboard system, flexibility is the ability to change the blackboard database implementation, the insertion/retrieval strategies, and the representation of blackboard objects without modifying knowledge sources or control code (i.e. maintaining an open system). Flexibility is important for two reasons.

- 1) The insertion/retrieval characteristics and the representation of blackboard objects are complex and hence subject to change as the application develops.
- 2) Even after an application prototype has been developed, the number of blackboard objects might vary from the original number as the application is used and enhanced.

The blackboard must be implemented so as to provide enough flexibility to permit these changes without necessitating changes to the knowledge sources (i.e. agents), the control code, or the blackboard database implementation machinery. Of equal importance, is the efficiency in the insertion and retrieval of blackboard objects which is normally achieved through improvements to the capability of control components.

In the past, most blackboard-based systems were developed new each time, implementing the blackboard model based on the criteria that appeared most appropriate for the particular application. Execution efficiency was the priority for some implementations with much effort expended on developing fast insertion and retrieval of objects on the blackboard. Making modifications to the blackboard structure or insertion/retrieval

methods were difficult due to the knowledge sources (i.e. agents) and control components being so tightly coupled to the underlying blackboard database. Applications that were designed for flexibility were developed on top of a general-purpose blackboard database retrieval facility such as a relational database system.

The blackboard system technology lends itself well as a tool for developing a cooperative problem-solving system and open system architecture within Farm Smart 2000. Several of these advantages are listed below.

- 1) Farm Smart 2000 is an integration of diverse, specialized agents, each representing a knowledge source. With the blackboard system's integration framework, existing agents as well as new agents can be developed with the most appropriate data representation that is best suited for maximizing their contribution to the total system, without compromising their integration.
- 2) Farm Smart 2000 has agents contributed from different sources and developers. The blackboard system approach can accommodate this disparity and maintain the agent's modularity and independence, enhancing the benefits of reusable software while providing flexible, dynamic control of problem-solving activities.
- 3) The blackboard system architecture provides a flexible and incremental approach to developing Farm Smart 2000 allowing ongoing development and integration of agents.

5.2.3 Providing Decision Support for Specific Individual Farming Systems

Farm managers are normally bombarded with information, but discover, after testing it against their particular situation, that much of the information is invalid for their specific farming operation. Thus the farm manager has a feeling of uncertainty for data and information or becomes skeptical, not knowing what to believe and what to ignore. Consequently, farm managers will seek out and validate, in their own mind, the sources of information and knowledge that are credible (i.e. goal #3, Section 4.3). These sources of credible information and knowledge might be, for example, persons, publications and/or the media. Whoever and/or whatever these sources are, farm managers analyze them as being able to provide a certain level of knowledge or information within a specific domain

as it pertains to their own farm. For a decision support system to obtain a high rating of credibility from the farm manager, its support and services must be:

- 1) accurate;
- 2) targeted to the specific farm manager's enterprise;
- 3) convincing (i.e. Be able to substantiate why it is making a particular recommendation.).

In order for an agricultural decision support system to fulfill requirements 1) and 2) above, on-farm databases must contain accurate and complete data. Specific recommendations can only result when there is sufficient on-farm data present to be used in the analysis. Hence, the most important requirement in developing a credible agricultural decision support system is implementing a sound record keeping system for collecting, validating, storing, retrieving and updating complete sets of on-farm data. Never has the old, but accurate, phrase "garbage in - garbage out" been more applicable. It makes little difference how "expert" the internal expert systems are or how "knowledgeable" the knowledge-bases are, if the data driving these internal components are not accurate, complete and relevant to the specific farm enterprise. Without a sound on-farm record keeping system, the agricultural decision support system will not be able to provide the required specific on-farm support, and will fail the farm manager's credibility test.

In fulfilling requirement 3) above, a successful decision support system must gain the confidence of the user. One of the downfalls of decision support systems, which has led to poor acceptance, is the inability of the decision support system to adequately explain and justify its recommendations and relate its reasoning process based on the farm manager's on-farm data and information [Barrett, 1992]. Farm Smart 2000 features the capability to view an explanation of the reasoning process that was used in arriving at the solution or recommendation provided to the user. Thus, if the user challenges the accuracy of the recommended solution, this information will justify the reasoning process and data used in solving the problem. This feature lends credibility to the decision support system, enabling the user to understand how the system derived its solution.

Establishing the necessary databases is necessary and very important to the development of an effective and credible decision support system. Data and information

support, service or "drive" the successful decision support system and hence data and information must be tightly integrated and consistent.

Not only on-farm data, but agronomic knowledge as well, are the driving forces behind Farm Smart 2000's problem-solving approach and planning solutions. Large amounts of agronomic knowledge are available within Farm Smart 2000 to complement the producer's on-farm data. This agronomic knowledge ranges from agro-climatic zones, to average yield for various crops, to threshold populations for pest management.

Establishing the necessary data- and knowledge- bases is very important to the development of an effective and credible decision support system. Farm Smart 2000 is "data driven" by SmartRKS which has an intuitive user interface built on the "folder and tab" concept which is extended throughout Farm Smart 2000 for a consistent "look and feel". Within this interface, input folders are used for data entry and output folders have been developed to summarize results that are focused at the producer's specific farming operations. With Farm Smart 2000 being "data driven", by internal system data and knowledge, combined with the producer's on-farm data (contained in SmartRKS), and in addition to being producer "goal driven", results are calculated that are specific to the individual producer's farming operation.

Another important aspect of Farm Smart 2000 and SmartRKS is that data are integrated throughout all modules contributing to data integrity and resulting in system credibility. This capability of "one-time data handling" provides the user with efficient data entry and minimizes the risk of inconsistent data so that once the user provides or creates data, it is always available regardless of the module in which it is used. For instance, the same crop data being utilized in the weed management module will also appear in the crop protection module and so forth. The database structures within SmartRKS also provide storage of management units, which are units of land within various fields that can be grouped together according to a common management practice. Not only does this capability enable specific data to be stored and related to small similar units of land for on-farm management purposes, it also makes Farm Smart 2000 adaptable to precision farming needs.

5.3 Development of an Open and Cooperative Problem-Solving System

The goals of cooperative multi-agent problem solving within an open system architecture for Farm Smart 2000 (i.e. goals 1 and 2, Section 4.3), required the development of a specialized blackboard system in combination with a unique global controller (i.e. Global Control Expert). For clarity, these were kept separate in the previous discussions (Sections 5.2.1 and 5.2.2) although in reality they are tightly coupled in Farm Smart 2000. A summary of this combination follows and then more specific details relating to their combined implementation.

The development of an open and cooperative multi-agent problem-solving system was achieved through the development of a blackboard system which provides a shared global database necessary for cooperative problem solving. The problem solving is performed by cooperating heterogeneous reusable agents which contain the knowledge required to solve the problems and are kept separate and independent. Various agents can be developed for any one application and each can use whatever technology best matches its purpose. The agents interact anonymously using the blackboard, a shared structured global database. The problem solving is directed by a Global Control Expert, a control shell, that is separate from the agents. Events (i.e. signals) form the communication between the Global Control Expert and the blackboard, and trigger agents. The blackboard model provides a number of important benefits including:

- modularity - agents can be developed independently by different collaborating institutions. Furthermore, since agents are reusable software systems or legacy software, they can be developed before or at the same time as the blackboard application itself;
- integration - agents can be implemented using various methodologies and approaches and then integrated to cooperatively solve problems;
- extendibility - the blackboard model provides an "open system" enabling new agents (i.e. knowledge sources/expert systems) to be easily added and existing agents to be deleted or replaced with enhanced versions as system specifications change;

- reusability - agents which provide generic expertise in one application can be reused in Farm Smart 2000.

5.3.1 The Blackboard Model/Database

A blackboard architecture consists of two major components: the blackboard database and a control mechanism [Lander et al., 1996]. The blackboard database differs from conventional databases in that it is specifically designed to provide efficient management of global information through a shared repository. This repository might contain, for example, problem specifications, partial and completed solutions, suggestions on how to proceed with a problem, information about the environment, and analysis of work completed or in-progress. With the blackboard's global visibility, agents need not duplicate shared information locally, avoiding the costs associated with duplicate data storage, as well as the problems associated with maintaining currency in local copies of shared data. Instead, sharing information via a blackboard is an efficient and effective method of ensuring that data are available to all agents that can use them, eliminating the need for information to be explicitly passed between agents as input/output since the agent can access it directly [Lander, 1995]. Furthermore, current parameter values are seen by all agents without update delays.

Within the problem-solving mode of the blackboard model, there are two methods for determining the next state in the solution process; search and recognition. At any solution step or state, the search method generates and evaluates the possible next states, whereas the recognition method simply knows what the next state should be through a pattern matching process that involves scanning the knowledge-base for knowledge that can be applied to a state [Jagannathan et al., 1989]. In the case of Farm Smart 2000, the recognition-oriented approach is taken. That is the knowledge-base is the Global Control Expert's Knowledge-base and the knowledge that can be applied to a state is either activation rules or conditional rules. Naturally, there may be several pieces of knowledge (i.e. rules) applicable for a given state and a decision is necessary to determine what knowledge (i.e. rules) to apply. This decision is also knowledge based, or in the case of Farm Smart 2000, rule based. In a recognition-oriented blackboard system, this decision, results in triggering either an activation rule or a conditional rule. The triggering of an

activation rule will fire and control a set of agents, based on the predetermined rules for a given situation, resulting in a change to the solution space on the blackboard. The triggering of a conditional rule will fire a predetermined rule set for a given situation resulting in a contribution to the solution space on the blackboard. In Farm Smart 2000, the activation rule sets, located in the Global Control Expert's Knowledge-base, trigger and control the reusable agents in their search for a solution. The conditional rule sets provide less complex problem-solving, where the superior intelligence of agents is not required.

5.3.1.1 Data Integrity

Data integrity refers to the accuracy or validity of data and a database management system ensures correctness and consistency of data through mechanisms for defining, managing, and controlling integrity constraints. These constraints are assertions that must be satisfied by objects in a blackboard database. The integrity constraints are typically classified as static constraints, which are conditions that relate to the state of the objects, and dynamic constraints, which are conditions that relate to state transitions of the objects. The most common types of constraints in a database management system are:

- domain constraints, which specify a set of acceptable values for an attribute;
- keys, which specify that the values of one or more attributes identify an object uniquely in a given set of objects;
- referential integrity, which specifies that if an object references another object, the latter must exist.

Blackboard databases apply these constraints except, being specialized forms of object-oriented databases, do not apply referential integrity as most blackboard databases do not allow the explicit deletion of objects. Instead, they use an internal garbage collection mechanism for determining which objects are not being referenced by other objects and therefore can be removed. Some additional specialized constraints are required in blackboard databases but because of the uniqueness of the databases, these integrity constraints are encoded internally in the system as part of the update methods associated with the objects.

Data integrity must be maintained when being accessed asynchronously by several expert systems. A common solution to this problem is the provision of transactional access

to shared databases where a sequence of operations exist for the data elements consisting of a start-transaction request and ending with either a commit- or an abort- transaction request [Ensor and Gabbe, 1988].

5.3.1.2 Information Efficiency

Efficiency is required in a blackboard database because not only do agents insert new contributions on the blackboard, but they also perform associative retrieval to find relevant information and data placed on the blackboard by other agents. This requirement, to find relevant information on the blackboard, is often not addressed in casual discussions of blackboard-based systems and yet associative retrieval is fundamental to the blackboard paradigm, where agents must be able to communicate anonymously among each other. Associative retrieval permits agents to look for meaningful information on the blackboard, when directed by the Global Control Expert, rather than receiving the information by appealing directly to other agents. In addition to anonymous communication, the time span from when objects are placed on the blackboard to when they are retrieved and used by another knowledge source, provides a global memory area for the agents.

The basic blackboard operations and past practices that have been used to implement associative retrieval in blackboard systems include:

- **Insertion** - Blackboard objects, once created, must be placed onto the blackboard with the creation of required pointers (i.e. locators) so that the object can be retrieved;
- **Merging** - When an object is placed on the blackboard, it may be important to determine if an identical object already exists on the blackboard, and if so, merge them into a single blackboard object that reflects both their hypotheses;
- **Retrieval** - The retrieval operation searches the blackboard for objects that satisfy a set of constraints specified in a retrieval pattern;
- **Deletion** - Deleting an object from the blackboard requires the manipulation of the locators (i.e. pointers) which point to it;
- **Repositioning** - When objects are added or deleted from the blackboard, consistency in the blackboard database must be maintained. Repositioning is

the management (i.e. modification) of object's locators (i.e. pointers) which are determined by the indexing attributes;

In the past, implementation of associative retrieval in blackboard systems has been characterized by three approaches to building blackboard applications.

- 1) An unstructured blackboard is a simple approach to building a blackboard application where each blackboard level is an unstructured list of the objects residing on that level. Agents push new objects onto the appropriate list in order to add them to the blackboard. Retrieval is implemented by the agents scanning the list for objects of interest.
- 2) The general-purpose kernel approach provides the agents and control component developers with a generic blackboard database facility. The facility supports blackboard object retrieval based on the object's attributes.
- 3) In the customized kernel approach, the appropriate retrieval strategy for each situation is custom-coded to avail of different retrieval strategies and to maximize retrieval speed. This is achieved through the development of an insertion/retrieval kernel that is customized to the situations that arise in a particular application.

There are many advantages to "publishing" data on shared blackboards for all agents to use. If data is shared through external files, that data is available only to the agent or agents that have access to those files. Furthermore, the data is available only in specific file formats, and thus, in order to use the information, the agent must know exactly what data it requires, where that data is located in the file, and the format of the data within the file. Sharing data through external files can be inefficient and can greatly restrict the use of the information, whereas data placed on a shared blackboard can be accessed and utilized by all agents through shared semantics and shared syntax.

5.3.2 Methods of Integrating Rules with Active Databases

One approach for achieving dynamic coordination is to generate rules from activity specifications that can be used for dynamic coordination of activities. These rules are more specifically termed event-condition-action rules [Diaz et al., 1991]. This dynamic

coordination of activities using event-condition-action rules is often referred to as active databases.

This methodology is used for achieving dynamic coordination from the Global Control Expert. Active databases have been defined as database systems that respond automatically, without user interaction, to events that are generated either internal or external to the system itself [Medeiros and Pfeffer, 1990]. Database system responses are declaratively expressed using event-condition-action rules proposed by Dayal et al. (1986). Event-condition-action rules are comprised of an event that triggers the rule, a condition describing a given situation, and an action to be performed if the condition is satisfied [Diaz et al., 1991]. Therefore, the system not only knows how to perform operations, but also when operations need to be performed, satisfying the requirements for goal-directed control.

A problem with incorporating rule management within active databases requires that there be a specific protocol for communications among the agents. Depending on the complexity of the problem being solved, this may not pose a major problem if an accurate coordination plan can be developed in a sequential manner. However, if the problem-solving is complex, as in Farm Smart 2000, where coordination requires knowledge of the actions of other agents and the ability to reason about the effect of those actions, then coordination must be derived dynamically necessitating that a dynamic coordination methodology be developed.

One focus of this research is to extend the blackboard model beyond its traditional role. Traditionally, the blackboard model supported the development of expert systems, whereby knowledge sources, which have been developed to work cooperatively, are combined to form a single expert system (i.e. agent) to solve a common problem. In contrast, Farm Smart 2000 builds on the blackboard model, enhancing it to support the design and implementation of a broader decision support system, comprising several heterogeneous reusable agents that know nothing about each other and using a Global Control Expert as the controlling mechanism.

The requirements, entities, and their interaction, to provide a rule manager in the context of an object oriented database system, were described in Section 4.2.7.2. Previous

approaches taken for incorporating rules into an object-oriented database management system basically fall within two methodologies:

- 1) rule specification at class definition time (i.e. parameterizing)
- 2) rule creation, activation, deactivation, and binding all performed at runtime.

The preferred programming language and desired system performance can influence the approach used. Previous methodologies for supporting active capability in an object-oriented model were disjoint, not considering the difference in the data models. Consequently, the functionality and seamlessness requirements of supporting rules in an object-oriented database were not fully satisfied.

In order to accommodate rules, in an object-oriented database management system, they must be classified into two main categories, class level rules and instance level rules. Class level rules are applicable to all instances of a class and instance level rules are applicable to specific instances. Regardless of the rules classification, they are treated as first class notifiable objects. A rule is defined by an event which triggers the rule, the condition which is evaluated when the rule is triggered, and the action which is executed if the condition is true. The main difficulty in implementing rules lies in supporting the condition and the action parts of the rule which are code in the application's language. One method for implementing rules in an object-oriented database system is to create a class for each rule and let the rule be an instance of that class, where the condition and action parts of the rule will be implemented as methods of the newly created class. A second approach is to create only one rule class and instantiate each rule as an instance of the rule class.

5.3.2.1 An Example Scenario Using Active Database Technology

Farm Smart 2000 is very complex in its problem-solving capabilities. However, the following scenario attempts to illustrate the feasibility of employing blackboard database technology coupled with rule management within active object-oriented database technology to provide the required cooperative problem-solving environment for Farm Smart 2000. The following examples of Farm Smart 2000 reusable agents are used:

- 1) CROPMAN, a crop management expert system, which focuses on crop production including crop selection and rotation, herbicide requirements, fertility, and potential plant disease [de Gooijer, in press].

- 2) A Stepwise Technology Adoption Risk Reduction Tool expert system called STARRT, which evaluates the economic impact of different cropping decisions [Duhaime et al., in press].
- 3) A planting and residue expert system called PARMS, which determines the level of success of planting a crop given the planting equipment, current production practices and conservation goals [Smith et al., 1995].

Even without considering other reusable agents in Farm Smart 2000, such as weed and insect agents, these selected agents should illustrate the interrelated complexity associated with integrating these three heterogeneous reusable agents in a cooperative problem-solving environment. A producer, during pre-growing season planning, requests Farm Smart 2000 to advise on the crop(s) to grow with the best economic net return. The agents being used in this example are similarly integrated into the blackboard framework as illustrated in Figure 5.3.

Once execution is initiated, the Global Control Expert takes control and queries the Global Control Knowledge-Base to determine that crop candidates need to be determined first and that CROPMAN must be triggered into action to establish the candidate crops. The Global Control Expert queries SmartRKS (the producer's on-farm database) and required data is placed on the blackboard. The event generator, via the Global Control Expert, generates an event, in the form of an event-condition-rule, which activates CROPMAN, if the condition is satisfied that required data is present on the blackboard. (The agent activation is stored in a queue of pending agent activations and at the start of each scheduling cycle, the control shell (i.e. Global Control Expert) executes the agent activation at the top of the queue and continues to do so until the queue is empty.) CROPMAN determines crop candidates that can be grown based on crop rotations, fertility, soil moisture, potential plant disease, herbicide residue and agro-climatic zone and places the results on the blackboard.

The Global Control Expert, using event-condition-action rules (i.e. agent coordination) and the Global Control Knowledge-Base, determines that economic analyses needs to be performed next and that STARRT is the agent required to perform this analysis. The event generator, through the Global Control Expert, generates another event of the

same form as previously, which in turn triggers STARRT, since the required data is available on the blackboard. STARRT determines the best crop(s) to grow on each field based on present market outlooks and input costs, and places its results on the blackboard. The above process is repeated and PARMS is initiated which uses the information on the blackboard, machinery inventory, and current levels of crop residue to determine that the crop(s) recommended cannot be seeded without equipment modifications due to the high level of crop residue present. PARMS places the applicable information on the blackboard including the costs for equipment modification. The Global Control Expert determines that a conflict has been encountered with the input costs that STARRT produced and the input costs that PARMS produced (i.e. the additional costs for machinery modification) and determines that the net return needs to be reevaluated given the cost of equipment modifications. The Global Control Expert determines that STARRT is required again and the procedure to activate STARRT is performed. If the Global Control Expert determines that the net return indicates the recommended crops are not cost effective, then CROPMAN is activated again with imposed crop constraints that are placed on the blackboard and accessed by CROPMAN during its execution. The process continues until a viable net return results with no conflicts.

Table 5.1 summarizes and maps the alternative techniques and methods that were evident in the literature review with the methods and techniques that were actually developed and implemented in Farm Smart 2000, and explains why they were implemented in particular.

Table 5.1: Map of Methods/Techniques Developed and Implemented Compared to Alternatives

Literature Review ALTERNATIVES	Method/Technique Developed/Implemented	Why
<p>Agents</p> <p>a) Establish a global structure containing information specifically about each agent to use for agent coordination.</p> <p>b) Develop all agents with inherent knowledge and information about all other agents such that agents interact directly with each other.</p>	<p>A Global Control Expert was developed similar to a). Note: b) defies the term "reusable agent" and software reuse is a goal of Farm Smart 2000.</p>	<p>To establish a method of collaboration (i.e. the development of a problem-solving plan) and coordination (i.e. execution of the plan) whether pre-defined or dynamically developed or both.</p>
<p>Problem Solving</p> <p>a) Distributed problem solving by dividing the problem-solving tasks among a number of cooperating agents.</p> <p>b) Multi-agent systems coordinate the knowledge, goals, and plans of autonomous agents to solve problems through their joint intelligent behavior.</p>	<p>Multiple agents cooperate with each other but not "jointly", but rather through a global control agent.</p>	<p>Each agent has separate individual goals and is capable of solving problems within its specific area of expertise providing an open system, where agents can be added or replaced. Task division would create unnecessary problem-solving complexity.</p>
<p>Complicated tasks requiring more resources or knowledge than is held by a single agent are decomposed.</p>	<p>Required additional knowledge is supplemented by "mini agents" constructed of rule-sets. These mini-agents span knowledge gaps among agents (e.g. gray areas)</p>	<p>The control agent's ability to coordinate problem-solving would be unmanageably complex creating excessive system overhead.</p>
<p>Meta-information should enable an agent to reason about the past, current, and future actions of other agents.</p>	<p>Agent meta-information is maintained and accessed by the Global Control Expert only.</p>	<p>Excessive overhead required to manage meta-information accessible by each individual agent, especially in an open system where agents are readily added and replaced.</p>

Agents' capabilities are captured in a capability data/knowledge-base so that agents are able to reflect their role and the roles of other agents involved in the cooperative problem solving.

Conflict Resolution

- agent's determine which data are relevant to them and which are to others
- understand each agent's identity and beliefs and manage their differences appropriately through a global conflict hierarchy
- backtracking to determine the roots of disparity
- utilize a third party arbitrator
- utilize a negotiation model which incorporates case-based reasoning
- utilize compromise negotiation which establishes a concession of requirements on the part of each agent in order to satisfy other agent's requirements
- utilize integrative negotiation which identifies the most important goals of each party to use in formulating a basis for a new solution
- have each agent share meta-information with other agents, assimilate shared information from other agents and then use this information to improve their local understanding of the global solution space to achieve global solutions
- utilize compromise negotiation, interactively revising values by sliding them along some dimension until a mutually agreeable position is found

Agent's capabilities are maintained in a meta-information/knowledge-base accessible by the Global Control Expert

The Global Control Expert utilizes weighted constraints from the agent meta-information/knowledge stored in the Global Control Knowledge-Base and knows the agents' minimum constraints, leaving the main task of identifying and compromising the predominant constraints.

To maintain open system architecture, only the Global Control Expert need be changed when an agent is added or replaced.

By establishing a set of conflict attributes as part of the meta-knowledge for each agent, and defining threshold values for these attributes, the Global Control Expert can reason as to when to try compromise negotiation.

<ul style="list-style-type: none"> utilize relaxation methods which are invoked when the problem-solving activity is believed to be over constrained resulting in little or no progress in solution building and is accomplished when one or more agents relax some requirement on a goal or solution 	<p>The Global Control Expert evaluates the weighted constraints of the agents and the constraints being imposed by the solution analysis</p>	<p>In so doing, the Global Control Expert is able to determine if relaxation is feasible.</p>
<p>Cooperative conflict resolution based on explicitly capturing conflict-resolution expertise and organizing it into a useful taxonomy of conflict classes each with an associated applicable conflict resolution strategy.</p>	<p>The Global Control Expert achieves a globally optimal solution, sacrificing individual agent's goals in the interest of increased global benefit.</p>	<p>By utilizing compromise and/or surrendering less important goals, mutually beneficial solutions can be reached.</p>
<p>Arranging conflict classes in the abstraction hierarchy so that the more general conflict classes are arranged near the top of the hierarchy and represent domain independent classes, and the more specific classes are near the bottom and relate to domain dependent classes.</p>	<p>Not used.</p>	<p>The complexity of Farm Smart 2000 does not permit all the different kinds of conflicts to be pre-established, and hence could not be arranged into such an abstraction hierarchy of conflict classes.</p>
<p>Retain the process and results of decisions (i.e. cases) and subsequently use them so solve future related conflicts (i.e. case-based reasoning).</p>	<ul style="list-style-type: none"> - Modified version implemented for short term retention of data/cases where legacy data is maintained on the "blackboard". - Modified version implemented for long term retention of data/cases where the solution of known conflicts are developed as "rule-of-thumb" in pre-determined (or dynamically created) rule-sets. 	<p>As experiences (i.e. rules) are accumulated, system performance improves providing more intelligent support as essentially the system learns from experience.</p>

Utilize conventional database methods.	Object-oriented model/database used instead, coupled with knowledge-base(s)	<ul style="list-style-type: none"> - the transaction model used in conventional databases is based on a paradigm that concurrent transactions "compete" for resources rather than "cooperating" to solve a specific problem - coupling knowledge-base and database technologies improves data management by using knowledge-base techniques to manage complex relationships among data and to perform deductive data processing - this merging also improves knowledge management by using database techniques to maintain the factual data imbedded in knowledge, therefore reducing the size and improving the extendibility and maintainability of knowledge-bases - enables active databases to be utilized so that system can dynamically respond to internally or externally generated events by using event-condition-action rules
Opportunistic Control	Not used.	<ul style="list-style-type: none"> - Since control is reevaluated after each operator execution with no representation of long-term goals or plans, this leads to fragmented activity (i.e. hard to see the forest for the trees). - Instead, specific goals are accomplished through specification of problem-solving sequences using workflow rule-base notation.

Blackboard-based Problem-solving Concept

A modified Blackboard System was developed integrating heterogeneous agents specifically, rather than “knowledge sources”, which blackboard systems were traditionally used for in building expert systems and utilizing a Global Control Expert (i.e. moderator). (As an analogy, and in contrast to the “blackboard” metaphor in Section 4.2.5, envision that a group of human experts are gathered around a large blackboard to cooperatively work together to solve a problem. The problem-solving session begins when the initial data and problem are described on the blackboard which is a specialized database. A moderator examines the information on the blackboard and determines, using a dynamic reasoning process, which expert can or should apply their expertise next to the developing solution. When sufficient information is available for any of the experts to make a contribution, the moderator triggers the appropriate expert into action, who in turn makes their contribution on the blackboard, allowing other experts, via the moderator's reasoning capability, to apply their expertise. This process of adding contributions and information to the blackboard database continues until the problem is solved.)

- required agents to be integrated in a cooperative problem-solving environment
- blackboard system offered independence of expertise in that each agent has specific expertise relating to the problem and can contribute to the problem totally independent of the other agents
- diversity in problem solving (i.e. each agent with their individual approach can make contributions)
- flexible representation (i.e. agents can contribute any type of information).
- event-based activation (i.e. agents react to external events)
- problem-solving control via the control shell (i.e. Global Control Expert)
- incremental solution building by ongoing contributions from agents
- serves as global “memory” for all agents by maintaining legacy data

5.4 Data Mining/Discovery System

This chapter has explained the methods and technology utilized in the development of Farm Smart 2000. With Farm Smart 2000 being a knowledge system it might be presumed that it incorporates data mining/knowledge discovery. This section clarifies why data mining/discovery systems are not utilized in the development of, or within, Farm Smart 2000. A data mining/discovery system or knowledge discovery system, as they are also called, is a system that "discovers" or "mines" knowledge, that it previously did not have, from a database. That is, the data was originally not implicit in the database or explicit in its representation of domain knowledge. Knowledge can be thought of as a relationship or pattern among data elements that is valuable with respect to a given domain. The knowledge discovery system itself contains components that work together to efficiently identify and extract interesting and useful patterns (i.e. knowledge) from data contained in real world databases. In this process, the knowledge discovery system issues database queries, via a database interface, to obtain the data of interest. However, the techniques (i.e. components) required for data mining/discovery involve more than just querying a database. The raw data selected from the database management system is processed using extraction algorithms which produce candidate patterns. These patterns are then evaluated in anticipation that some will be interesting and useful discoveries. The results are presented to the user and also stored in the system's knowledge-base to support subsequent discoveries.

Many complicated interrelations exist among agronomic data and farming entities within conservation and diversified cropping farming systems, and consequently within Farm Smart 2000. Currently, very little agronomic data is collected electronically, but rather technicians and scientists usually collect only the data required from their research plots due to the expense of collecting superfluous data. Individual agronomic datasets/databases can rarely be combined, again because of the complex interrelationships that exist. The knowledge utilized within Farm Smart 2000 came from credible sources (e.g. scientific publications) where results have been already synthesized from collected data. If data were to be mined, they would need to be thoroughly analyzed to determine if and how they relate to the knowledge being produced. This would require multiple human

experts each specializing in a different field of agronomy. Hence, agronomic data mining/discovery systems are not currently cost effective to develop for limited datasets/databases. In the future when on-line agronomic data collection is more prevalent, possibly with precision farming and agricultural satellite imagery analysis, data mining/discovery systems might be more cost effective.

Chapter 6: The Use and Design of Farm Smart 2000

With the reduction of government extension specialists and ever increasing competition in crop production, producers are approaching agricultural retailers or private consultants more often for information, answers, and recommendations pertaining to their crop production problems. Furthermore, farm managers are being forced to seek out more answers and information on their own. Tier-3 of Farm Smart 2000, whether it be located on the computer of the farm manager, retailer or consultant, is able to support the decision-making process by solving interrelated conservation farming problems.

The following sections describe how Farm Smart 2000 was tested and validated and discusses how Farm Smart 2000 will be used as a technology transfer tool. A description of how to use Farm Smart 2000 is provided indicating the required input data and how the results are displayed. Farm Smart 2000's tier-3 components, rule management, and its process of evaluation are described. The fulfillment of the general user requirements and specifications are discussed in terms of problems, solutions, compromises and limitations. This chapter concludes with a summary of Farm Smart 2000's uniqueness and the non-existence of similar applications.

6.1 Testing and Validating Farm Smart 2000

There are basically five phases to be considered in the software development process; problem analysis, solution development, implementation, testing and program maintenance. Inevitably some testing is performed as part of the solution development and implementation phases as programmers mentally test code as it is produced and simulate the execution of modules prior to any formal testing. With Farm Smart 2000 being an open system (and its prototypes attempting to be), the development process was iterative with executable versions being tested internally by the working groups, then enhancements and modifications recommended by the Steering Committee were made to the system, followed by the internal testing phase again. This iterative process continued until Farm Smart 2000 reached a level of problem-solving capability that satisfied the Steering Committee.

SmartRKS was developed before the completion of Farm Smart 2000 as it contains many of the required databases for Farm Smart 2000, and hence, it was imperative that the database structures be developed initially. As a result, an external beta version of SmartRKS was completed first which was distributed to approximately 85 producers throughout Saskatchewan and a few in Alberta and Manitoba. Throughout December 1998, each of these producers participated in one of eight half-day workshops held in their area regarding SmartRKS's capabilities and how to use them.

The internal beta version of Farm Smart 2000 is being tested by the Saskatoon Working Group who are composing a list of problems (e.g. programming errors) and possible further enhancements. A producer, who was on the original PARI DSS Steering Committee, has also evaluated this internal beta version of Farm Smart 2000. The result of his evaluation is Appendix B. Another workshop will be held where Farm Smart 2000 internal beta version will be introduced to an extended PARI DSS Steering Committee including other various experts. These experts will test and validate Farm Smart 2000 working from this list and adding to it. It is expected that during this validation, these experts will compare Farm Smart 2000's results with results from their own research. These are the metrics evaluators choose to use since a base set of test results is impossible to produce with Farm Smart 2000's capability to dynamically alter the problem-solving process. It will then be decided what further enhancements will be developed, then these enhancements will be implemented, and an external beta version of Farm Smart 2000 will be produced. After this external beta version is tested by the working groups, it will be distributed to the same producers who received SmartRKS earlier. By this time, these producers will have populated SmartRKS with their on-farm data and will be ready to test and evaluate the entire Farm Smart 2000 system.

The producers' error reports and evaluations will be reviewed and final modifications will be made to produce a Farm Smart 2000 Version 1.0 which will then undergo additional in-house testing and validation by the working groups before being offered (i.e. marketed) for wide distribution.

6.2 Farm Smart 2000 as a Technology Transfer Tool

Farm Smart 2000 provides decision support for conservation farming systems by advising managers on crop rotation, disease, fertility, weed management, appropriate machinery for managing residue, and applicable production economics. Not only can Farm Smart 2000 provide pertinent and specific decision-making support, but also supply detailed explanations of reasoning procedures to rationalize the recommendations specified, contributing to an increased level of credibility.

Farm Smart 2000 is an ongoing system for transferring technology from researchers and scientists to their clients, producers and farm managers. With Farm Smart 2000 being an open system, frequent updates to pesticide registrations, updates to crop varieties (e.g. more pulse crops), and the update of agents (e.g. addition of rangeland management and manure management), can all be accommodated. Results from research projects (e.g. thresholds and agronomic value results) can be integrated into existing databases, rule sets or look-up tables.

Farm Smart 2000 will be distributed by an industry partner with the intention that the innovative producers will obtain it first, with the trickle-down effect resulting in their neighbors acquiring it. The dissemination of Farm Smart 2000 will be aided by government and private industry crop specialists and service agents using Farm Smart 2000 in consultation and support of their clients' (e.g. producers) crop production problems and questions.

Farm Smart 2000 will also have some positive side effects in that it will provide:

- a permanent record of expertise (i.e. expert systems do not retire or relocate);
- access by less experienced persons to high level expertise (whether they are extension consultants, farm managers, or students);
- an educational tool for training people (university students, extension consultants and farm managers);
- assistance in structuring (i.e. formalizing) the knowledge of experts
- easy low cost knowledge replication (human experts require expensive training and education);

- greater distribution of expert knowledge (right to the farm manager's farm or agricultural service agency); and,
- unbiased accurate meaningful support for farm decisions through the integration of multiple agents in a cooperative problem-solving environment.

All people involved in technology transfer, whether it be public or private, are overburdened with information. Decision support systems, such as Farm Smart 2000, can assist in the quick retrieval of information and acquisition of knowledge.

Until decision support and expert system technology are successfully and widely transferred by the "innovative" farm managers and extension personnel to the general body of farmers, Farm Smart 2000 will be most frequently used by the innovative farmers and "front-line" technology transfer people. These people will include university and government extension personnel, commodity and conservation group extension people, as well as private sector distributors and retailers, who many farmers rely on as information providers. Expert systems such as the Crop Protection Planner are not only being used by producers, but are frequently used by both farm service retailers and extension agents as well.

6.3 SmartRKS: Smart Record Keeping System



As discussed in previous sections, the accuracy, credibility, and ultimate success of a decision support system is determined by the availability of sufficient site-specific data to use in processing, analyzing, and reasoning. SmartRKS was developed in response to the need for a record keeping system that would collect, store and retrieve detailed on-farm data for use by the problem-solving modules of Farm Smart 2000. Although a suitable existing record keeping system that would fulfill the requirements of Farm Smart 2000 was sought for reuse, none was found that met the requirements. The ability to provide decision support which is specific to the producer's farming operation is dependent upon the system's ability to access specific on-farm data as well as external data sources. Reliable farm and field records, detailing the history of farming operations on each field, are essential to ensure that the appropriate recommendations are made. SmartRKS provides for the collection and presentation of comprehensive field-level production information.

A successful record keeping and planning tool must allow the producer to record the past, evaluate the present, and plan for the future. SmartRKS with Farm Smart 2000, provide farm managers this valuable management capability. Production records are not just a historical record, but rather SmartRKS allows detailed field records to be managed effectively, and in combination with external data such as weather, crop variety, and crop inputs, it provides benefit to the producer through Farm Smart 2000 planning capabilities.

SmartRKS supports four main types of data: land, inventory (inputs and equipment), events, and system data. Land data contains information about farms, field groups, fields, and management units. Inventory data contains information about supplies on hand (or easily accessible) and equipment. Event data contains information about events that occur in a farming operation. System data is data that are used in the SmartRKS, but not editable by the user.

6.3.1 Using SmartRKS

SmartRKS enables farm managers to store and view information pertaining to their farms and farming operations including land, equipment, inputs and field operations. With specific numerical information being entered, such as quantities, measurements, and temperatures, SmartRKS enables the user, via the “preferences” menu, to select the units which they prefer whether they be in metric or SI format such as hectares, litres or kilograms, or in Imperial format such as acres, gallons and pounds. Users can change their preferences on a case-by-case basis if desired. To protect against loss of data, an auto-save “check-off” box is available in the preferences to enable or disable the automatic saving of newly entered or modified records. When auto-save is disabled, the user is prompted to save records after new entries or modifications.

Entering farm and field information is accomplished using a tree-like scope selector with collapse  and expand  marks. The current year is always displayed enabling the user to confirm that accurate farm and field data are being entered for the correct year as illustrated in Figure 6.1. The user is prompted to correct any information that SmartRKS does not recognize as valid. The user may make changes to a record’s data, but until that record is saved, the changes are only temporary. The user can click the cancel button and the record data will revert to the last saved data at the current position. Fields can be

combined with each other to form field groups or further subdivided into management units (See Figure 6.2). Management units are units of land within various fields that can be grouped according to a common management practice. Hence, not only does this capability

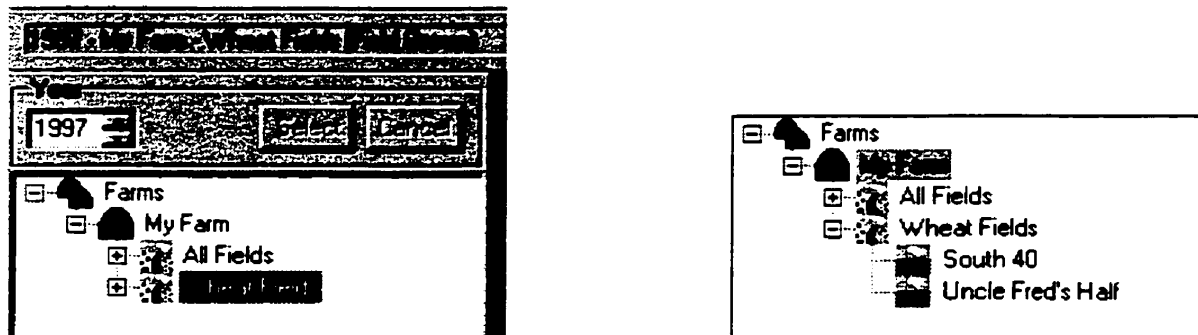


Figure 6.1: Scope Selector Showing Displayed Year and “Wheat Fields” Being Expanded



Farm: The farming operation in its entirety

In this example, the farm consists of 2 adjacent sections in the south (i.e. 8 quarters) with an additional quarter section on the northern boundary which is connected to another half section to the northeast.



Field: A individually managed parcel of land

The quarter section shown is managed as a separate field within the farm. Note: A field can belong to any number of field groups (e.g. “Barley Fields” group and “Malting Barley Fields” group).



Field Group: Two or more fields that are managed the same or similarly

The three fields shown are defined as a field group as they are managed similarly in the current year (e.g. “Barley Fields” group). Note: Records entered for a field group are stored as records for each individual field in the group. Hence, records are maintained at the field level when a group is changed or deleted (e.g. from year to year).



Management Unit: An area within a field that is managed differently from the remainder of the field.

The west 80-acre portion of the southwest field as shown is a management unit, as it is managed differently from the rest of the field. For example, this 80 acres was treated with an insecticide to eradicate migrating grasshoppers from the west field boundary. It could then be defined as a management unit within this field, allowing unique records related to the insect infestation and the particulars of the insecticide application to be entered for that portion of the field. Note: A field can have numerous management units. Management units can be further subdivided.

Figure 6.2: Land Definitions and Examples

enable specific data to be stored and related to small similar units of land for on-farm management purposes, but also makes Farm Smart 2000 "precision farming friendly". SmartRKS includes an on-line glossary of terms used in the program which is available from the help menu.

Most data in the SmartRKS database are stored by its scope (i.e. land/year combination). During the execution of SmartRKS, there exists a "current scope" at all times. This scope is global to SmartRKS in that it is not possible to view data from two different scopes at one time. The scope controls which data is accessible at any one time and where newly entered data is stored. The current scope can be changed at any time with the scope selection tools (i.e. the scope selection bar and/or pop-up dialog). The current scope is displayed on screen via the scope selection bar and/or the scope display bar shown in Figure 6.1. Each item appearing in the scope selector tree has a right-click menu (See Figure 6.3) corresponding to it to simplify the management of land and reporting features.

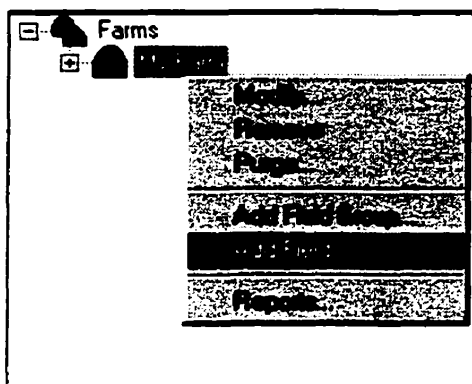



Figure 6.3: Scope Selector Showing Pop-up Dialog

6.3.1.1 Toolbars

SmartRKS has several toolbars which contain icons or buttons that are small pictures indicating their purpose. Most of the commands listed in SmartRKS's menus have a corresponding button so that clicking on a button is equivalent to opening a menu and selecting the desired menu command. SmartRKS's toolbars are movable and dockable so that they can be moved to convenient locations on the screen and then when moved back into the vicinity of their default location, they re-position themselves, back into their default positions. Each toolbar button also has "tool tips" which are small windows of information that describe the function of the button and are activated when the mouse cursor is placed

over any toolbar button. For example, moving the mouse cursor over the  button displays the “Equipment Records” tool tip message.






The main toolbar contains the following buttons, which correspond to choices from within the Records, Reports, and Help menus.

-  **Land Records**
-  **Input Records**
-  **Equipment Records**
-  **Event Records**





-  **Report Wizard**
-  **Open Saved Report**


-  **About SmartRKS**


The navigation toolbar is comprised of the following buttons, which are used for folder navigation. These buttons correspond to choices on the navigation menu.

-  **Go to First Record.**
-  **Go to Previous Record.**
-  **Go to Next Record.**
-  **Go to Last Record.**
-  **Create New Record.**

The record maintenance toolbar contains the following buttons, which correspond to the record maintenance menu.

-  **Save Record.** Saves the current record to disk.
-  **Cancel Changes.** Cancels any unsaved changes.
-  **Delete.** Deletes the current record for the current scope.
-  **Apply Record.** Makes a copy of the current record and allows the user to select the scope to which the copy should be applied. (Note: This capability can expedite data entry in the initial stages.)

 **Notes.** Provides an area in which the user can add a note to any record, in any folder, or view a previously added note. The word NOTES appears in the Status Bar whenever a record with a “note” is displayed, that is, if a note has been added to the current record. A note can contain additional information about equipment, inputs or field operations.

 **New Window.** Opens a new window so the user can view multiple records at the same time.

To start entering on-farm records the user would begin by entering farm information by selecting “Land” from the “Records” menu and would establish the farming enterprise to which all other data will be linked. More than one farming enterprise can be established, for instance in the case of a father and son or daughter farming together but with their own land holdings. After establishing the farm enterprise(s), the user would enter field information, again by selecting “Land” from the “Records” menu, and then “adding a field”. From within the “Land” function, the user can choose the following buttons (i.e. functions):

- 1) Add farm (e.g. year acquired (spin wheel¹⁵ available), year disposed (spin wheel available), farm name, last name, first name, email address, postal address, town/city, province (drop down list box¹⁶ available), postal code, primary phone, primary fax);
- 2) Add field (e.g. year acquired (spin wheel available), year disposed (spin wheel available), field name, land use of field (drop down list box available), field size, land value, soil zone (drop down list box available), Legal Location: quarter, section, township, range, meridian; soil name, soil texture (drop down list box available), landform (drop down list box available), soil polygon number (if known), slope, slope length, soil organic matter, soil pH);

¹⁵ A spin wheel is similar to a mechanical spin wheel (e.g. found on the back of electronic devices like computer devices to select the unit/channel number) and is implemented with an up and down arrow adjacent to each other with the up arrow “spinning” the number forward and the down arrow “spinning” the number backward.

¹⁶ A drop down list box is implemented with a down arrow which “drops” a list of data items to select from. If for instance, an implement name is required as data, the drop list for this entry will consist of implements that the producer has entered in the equipment folder (i.e. one-time-data entry). Or, if a herbicide name is required as input, the drop list will contain herbicides that were entered in the inputs function.

- 3) Add field group (e.g. year defined (spin wheel available), final year (spin wheel available), field group name, fields in group (from two lists, arrows permit the user to move fields from “fields not in group” to “fields in group” or back, if the user changes their mind));
- 4) Add management unit (e.g. year defined (spin wheel available), final year (spin wheel available), management unit name, soil name (brought forward from parent field, can be modified), soil texture (brought forward from parent field, can be modified, drop down list box available), landform (brought forward from parent field, can be modified, drop down list box available), soil polygon number (brought forward from parent field, can be modified), slope (brought forward from parent field, can be modified), slope length (brought forward from parent field, can be modified), soil organic matter (brought forward from parent field, can be modified), soil pH (brought forward from parent field, can be modified), percent of field (entered directly or by sliding an indicator on a ruler which shows the percent and calculates the actual number of acres) Note: as management units are defined within each field they are displayed in another window where the user can see the percentage of field each management unit consumes, the approximate acres each consumes, the total percentage all the management units consume, the approximate acres all the management units consume, and provides for editing and deleting management units before being committed to the database);
- 5) Remove (e.g. farm, field or field group, management unit);
- 6) Modify (e.g. farm, field or field group, management unit (including subdividing management unit): the same windows are displayed as used for “adding” above;
- 7) Purge (e.g. delete all records for the currently-displayed folder item for a range of dates. For example, the user might wish to purge (i.e. delete) old long term debt records that are no longer relevant or required);
- 8) Reports.

As mentioned, SmartRKS uses a folder and tab method of entering and storing records. Folder windows show tabs which are labeled with categories of records that can be kept in the folder. Each tab can contain multiple records which can be viewed by selecting items on the navigation menu or by using buttons on the navigation toolbar. The “Records” menu provides for the choice of three other types of records to be entered which are folder based and include:

1) Equipment Records Folder which has six tabs and permits the user to enter and store information regarding:

- Farm buildings (e.g. year acquired (spin wheel available), year disposed (spin wheel available), building name, building purpose, building value, year built (spin wheel available), total repair cost, depreciation (spin wheel available));
- Tractor equipment (e.g. year acquired (spin wheel available), year disposed (spin wheel available), tractor name and model, tractor value, horsepower (spin wheel available), total repair cost, depreciation (spin wheel available), hours of use (spin wheel available));
- Tillage equipment (e.g. year acquired (spin wheel available), year disposed (spin wheel available), machine name and model, machine type (spin wheel available), machine value, residue reduction, machine width (spin wheel available), fuel usage, total repair cost, depreciation (spin wheel available), hours of use (spin wheel available));
- Seeding equipment (e.g. year acquired (spin wheel available), year disposed (spin wheel available), machine name and model, machine type (drop down list box available), machine value, coulter present (check-off box), opener type (drop down list box available), number of ranks (spin wheel available), rank spacing (spin wheel available), row spacing, seed row width (spin wheel available), separation of fertilizer and seed, vertical clearance (spin wheel available), machine width, fuel usage, total repair cost, depreciation (spin wheel available), hours of use (spin wheel available), indicates: seed bed utilization);

- Spraying equipment (e.g. year acquired (spin wheel available), year disposed (spin wheel available), machine name and model, machine value, machine width (spin wheel available), fuel usage, total repair cost, depreciation (spin wheel available), hours of use (spin wheel available));
 - Harvest equipment (e.g. year acquired (spin wheel available), year disposed (spin wheel available), machine name and model, machine type (drop down list box available), machine type (if combine, combine type (drop down list box available), machine value, spread width (spin wheel available), cutting width (spin wheel available), fuel usage, total repair cost, depreciation (spin wheel available), hours of use (spin wheel available), Residue Equipment (check-off boxes): straw spreader, chaff spreader, straw chopper);
 - Other equipment (e.g. baler, harrow; year acquired (spin wheel available), year disposed (spin wheel available), machine name and model, machine purpose, value, fuel usage, total repair cost, depreciation (spin wheel available), hours of use (spin wheel available)).
- 2) Input Records Folder which has 7 tabs and permits the user to enter and store information regarding:
- Fungicide inputs (e.g. year acquired (spin wheel available), year disposed (spin wheel available), fungicide name (drop down list box available), cost, application units (drop down list box available));
 - Herbicide inputs (e.g. year acquired (spin wheel available), year disposed (spin wheel available), herbicide name (drop down list box available), herbicide group, cost for up to three herbicides);
 - Insecticide inputs (e.g. year acquired (spin wheel available), year disposed (spin wheel available), insecticide name (drop down list box available), cost);
 - Fuel inputs (e.g. year acquired (spin wheel available), year disposed (spin wheel available), fuel name, cost);

- Fertilizer inputs (e.g. year acquired (spin wheel available), year disposed (spin wheel available), fertilizer name, fertilizer cost, nitrogen source (drop down list box available), Formulation %: Nitrogen, Phosphorous, Potassium, Sulfur, Chloride, Copper, Zinc, Manganese, Iron, Boron);
 - Seed inputs (e.g. year acquired (spin wheel available), year disposed (spin wheel available), Seed: crop name (drop down list box available), crop variety (drop down list box available), seed name, seed storage moisture, germination, 1000-kernel weight (spin wheel available), seed cost, seed treatment applied, Inoculant: inoculant name, inoculant rate);
 - Other inputs (e.g. year acquired (spin wheel available), year disposed (spin wheel available), input name (e.g. liquid manure), unit cost).
- 3) Field Operation Records Folder which has 22 tabs and permits the user to enter and store information regarding:
- Custom Work, as it pertains to each field (e.g. date (date selector¹⁷ available), contract description, contract details: amount worked, rate, calculates: contract amount);
 - Disease, as it pertains to each field (e.g. disease name (drop down list box available), date first symptom (date selector button available), plant part injured, general appearance, disease distribution (drop down list box available);
 - Disease Control, as it pertains to each field (e.g. Disease: disease name (drop down list box and data button¹⁸ available), control level (drop down list box available), fungicide: calculate button¹⁹, fungicide applied (drop down list box and data selector button available), application date

¹⁷ The date selector is a button beside the date entry point which looks like a month on a calendar and when clicked presents a month of the calendar from which the date can be selected and month and year can be selected using spin wheels located below the month. Using the date selector ensures proper format of date entry.

¹⁸ The data button enables the user to directly access the data being referenced so as to add a record or modify a record. For instance, if the user is accessing the disease control folder and realizes that a another disease (record) needs to be added to the disease folder, the data button beside the disease name entry can be used to proceed directly to the disease folder to make the addition.

¹⁹ The calculate button provides a calculator function to the user with which to perform calculations. This is sometimes accomplished by an agent being automatically accessed/invoked.

(date selector button available), amount treated (spin wheel available), application rate, Calculated: fungicide cost, duration of activity (spin wheel available));

- Fertilizing, as it pertains to each field (e.g. fertilizer name (drop down list box and data button available), implement name (drop down list box available), application method (drop down list box available), application date (date selector button available), area treated, application rate, soil moisture, duration of activity (spin wheel available), indicates: components applied in terms of lb/ac of Nitrogen, Phosphorus, Potassium, Sulfur, Chloride, copper, Zinc, Manganese, Iron, Boron);
- Fixed Costs, as it pertains to the farm (e.g. office costs, farm insurance, utilities);
- Hail/Crop Insurance, as it pertains to each field (e.g. type of policy, insurance company, insurance cost, payouts);
- Harvest, as it pertains to each field including all management units, unless indicated otherwise (e.g. Equipment: implement name (drop down list box and data selector button available), residue management check-off box, Costs: drying costs, trucking costs, Harvest: crop name (drop down list box available), grade of crop (drop down list box available), moisture, dockage, cutting height (spin wheel available), harvest date (date selector button available), acres harvested, yield, total yield is indicated, duration of activity (spin wheel available);
- Hired Labour, as it pertains to the farm (e.g. employee name, Labour Details: amount worked, hourly rate, Calculated: labour cost);
- Insect, as it pertains to each field including all management units, unless indicated otherwise (e.g. Insect Infestation: insect name (drop down list box available), inspection date (date selector button available), growth stage (drop down list box available), feed source (drop down list box available), location on plant (drop down list box available), distribution

in field (drop down list box available), density (drop down list box available));

- Insect Control, as it pertains to each field including all management units, unless indicated otherwise (e.g. Insect Infestation: insect name (drop down list box and data selector button available), control level (drop down list box available), Insecticide: calculate button, insecticide applied (drop down list box and data selector button available), application date (date selector button available), area treated (spin wheel available) application rate, Calculated: insecticide cost, duration of activity (spin wheel available));
- Long Term Debt, as it pertains to farm (e.g. loan name, bank or credit union, loan start year (spin wheel available), amount of loan, length of loan (spin wheel available), interest rate, payments/year (spin wheel available), Calculated: payment amount, principal payment, interest payment);
- Market, as it pertains to farm (e.g. crop name (drop down list box available), crop grade (drop down list box available), date of sale (date selector button available), amount sold, price, buyer, delivery point, Calculated: total sale amount, trucking costs, storage costs, commodity checkoffs; Additional Calculated: total costs, net sale amount);
- Operating Loan, as it pertains to farm (e.g. loan name, bank or credit union, amount of loan, length of loan (spin wheel available), interest rate, payments/year (spin wheel available), Calculated: payment amount);
- Other, as it pertains to each field including all management units, unless indicated otherwise (e.g. event data (date selector button available), area (spin wheel available), cost, duration of activity (spin wheel available));
- Property Tax, as it pertains to each field (e.g. assessed value, property taxes);
- Rent, as it pertains to each field (e.g. rent description, rent amount, date rent paid (date selector button available));

- Seeding, as it pertains to each field including all management units, unless indicated otherwise (e.g. crop name (drop down list box and data selector button available), underseeded crop (drop down list box available), seed name (drop down list box available), implement name (drop down list box and data selector button available), packing method (drop down list box available), seeding date (date selector button available), area seeded (spin wheel available), seeding depth, seeding speed, seeding rate, yield target, soil moisture, soil temperature, emergence date (date selector button available), duration of activity (spin wheel available), Simultaneous Fertilizing: - Fertilizer 1 Application: use this application check-off, fertilizer name (drop down list box available), application method (drop down list box available), Calculated: application rate, Fertilizer 2 Application: use this application check-off, fertilizer name (drop down list box available), application method (drop down list box available), Calculated: application rate);
- Soil Test, as it pertains to each field including all management units, unless indicated otherwise (e.g. Sample: sampling date (date selector button available), test units used (drop down list box available), sampling technique (drop down list box available), soil test lab (drop down list box available), Nutrient Levels in Soil: sampling depth (up to 3 entries with spin wheels available), nutrients to be applied, for each of up to 3 depths and nutrients to be applied, values for: Nitrogen, Phosphorous, Potassium, Sulfur, Chloride, Copper, Zinc, Manganese, Iron, Boron);
- Tillage, as it pertains to each field including all management units, unless indicated otherwise (e.g. implement name (drop down list box and data selector button available), date of operation (date selector button available), total area tilled (spin wheel available), duration of activity (spin wheel available));

- Weather, as it pertains to farm (e.g. event description, date of event (date selector button available), amount);
- Weed, as it pertains to each field including all management units, unless indicated otherwise (e.g. weed name (drop down list box available), emergence date (date selector button available), weed stage type (drop down list box available), weed stage, density (drop down list box available));
- Weed Control, as it pertains to each field including all management units, unless indicated otherwise (e.g. for each of up to four weeds: Weed - weed name (drop down list box and data selector button available), control level (drop down list box available), Herbicide - herbicide applied (drop down list box and data selector button available), application date (date selector button available), area treated (spin wheel available), application 1 rate, application 2 rate, application 3 rate, Calculated: herbicide cost, duration of activity (spin wheel available), Calculate: accesses Crop Protection Planner agent, Crop: stage type (drop down list box available), crop stage).

6.3.1.2 Generating SmartRKS Reports

SmartRKS provides a comprehensive reporting facility called a report wizard which allows the user to select the type and contents of the report and then view or print the resultant report. Reports are based on the four categories of records described; land records, equipment records, input records, and field operation records, and are displayed in report windows of which several may be open at any time. In addition, special reports can be prepared for several items that are derived from two or more of these categories.

The Report Wizard helps the user select items for a report. The wizard asks the user to select these items one at a time, and continues to present logical options based on previous user choices.

The report wizard contains several options which allow the user to customize the report. These options include:







- Report Type - allows the user to select the type of report to be generated;

- **Special Report Type** - allows the user to select the type of special report to be generated;
- **Year** - allows the user to specify the inclusive range of years to be included in the report;
- **Scope Depth** - allows the user to specify the deepest scope level to be used in the report;
- **Scope** - allows the user to specify the scope used to filter land selection. If the deepest scope level is a farm scope, this option is not selectable. Combination selections are available depending on the scope level;
- **Land** - allows the user to specify the specific land items to be included in the report being generated. The higher-level scope selection performed previously is used to filter the available land items;
- **Events** - allows the user to select the event types that will be included when generating the report. The list of available events is dependent on the deepest scope level. Farm events are listed only when the scope is at the farm level. Similarly, field events are listed only when the scope is at either the field group or field level. This page also appears for the special-complete cost report, but the selections available differ slightly. The selection list for this report contains events, inputs and equipment, although not all events, inputs or equipment have costs related to them, and therefore only the relevant ones are listed.
- **Inputs** - allows the user to select the input types to be included when generating the report;
- **Equipment** - allows the user to select the equipment types that will be included when generating the report;
- **Grouping** - allows the user to specify the grouping for the report. This option appears only when two or more grouping items (i.e. year, land, or events/inputs/equipment) form the report;
- **Detail** - allows the user to specify the amount of detail to include in the report.

6.3.1.2.1 The Report Module

A separate report module provides the actual report display and printing functionality. This module contains a number of methods for creating a multi-column report with multiple sections, with each section having different column layouts. As well, it allows different font sizes and text effects to be used. An important feature of the reporting module is its “auto-sizing” capability. It is designed to automatically reduce font sizes until the report can fit horizontally on the screen. Thereby, the user does not need to “pan” to see the entire report. This feature is equally important when printing, as it ensures that the report can fit on one page width. Reports can also be saved and later reloaded by the user.

The report toolbar appears when a report is displayed and contains the following buttons that are equivalent to some of the choices on the reports menu.

-  **Open a Saved Report**
-  **Save this Report**
-  **Print this Report**
-  **Print Preview**
-  **Return to Report Wizard**
-  **Refresh this Report.** Regenerates the displayed report using current data.

6.3.2 Time Ranges

A primary design goal of SmartRKS is to ensure that data entered for one crop year do not affect data previously entered for different crop years. To accomplish this, each land or inventory item that is added to SmartRKS has a year-acquired data field and a year-disposed data field. The values in these data fields are entered when the item is added to the database. As an example, there might be a field that was purchased in 1986 and sold in 1994. Its year-acquired value would be 1986, and its year-disposed value would be 1994.

If some of the field’s data changes during this time, additional records might be created for the field. Each record has two additional data fields: first-year valid and last-year valid, indicating the range of years for which the data in the data field is valid. There may be three records for one field, each of which is valid for a different range of years.

These year ranges will never overlap, and will always cover the entire range between year acquired and year disposed.

When a year is specified in the scope for a folder, only records that are valid in that year will be accessible. It is possible to browse records with an unspecified year. In this case, all records will be accessible, and the agent is responsible for handling the multiple records.

6.3.3 Data Validation

SmartRKS provides validation functionality for all data in its database. When an agent attempts to modify the database, SmartRKS performs internal validation testing on the data. If the testing fails, then the data is not added, and the routine will fail. However, this method of notification has an “all-or-nothing” approach in that either all of the data passes the validation testing, or none of it does. Thus, it is not known exactly which piece of data caused the offense. To pinpoint offending data, SmartRKS provides validation routines that can be used for individual data fields. The agent is responsible for testing each field of a folder to determine its validity status, unless the “all-or-nothing” approach is satisfactory.

6.3.4 SmartRKS Technical Details

As described, SmartRKS is a self contained record keeping system capable of storing and retrieving on-farm data, performing calculations and summations, and reporting results. As such, it can be used as a stand-alone application and has been for beta testing as described in Section 6.1. For Farm Smart 2000 tier-3, (i.e. Crop Planning) to analyze scenarios, there must be a minimal set of data stored in SmartRKS to be used in calculations and analyses.

The SmartRKS database consists of a number of database tables in various formats (e.g. Paradox and dBase). Access to these database tables, for all agents including the SmartRKS application, is provided by an Application Programming Interface (API) that consists of a set of functions in a Windows Dynamic Link Library (DLL) as illustrated in Figure 6.4. In addition, there is a SmartRKS administration application that is used by system administrators to enter, modify, and delete internal system data in SmartRKS.

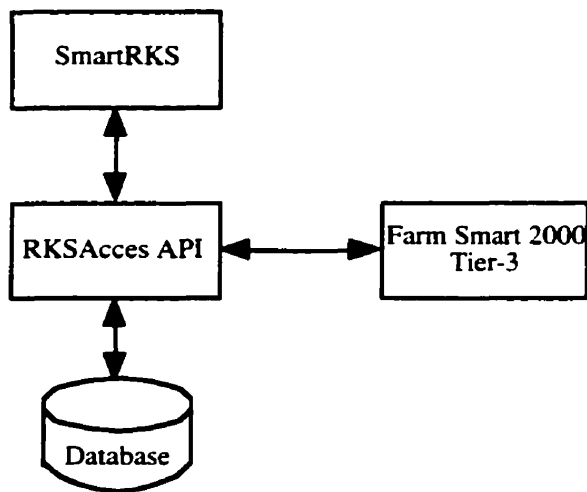


Figure 6.4: Access to SmartRKS Database

6.4 Farm Smart 2000 Tier-3

As indicated in Section 5.1, the focus of this dissertation is Farm Smart 2000's tier-3, Interrelated Decision Support for Crop Planning, and thus any mention in the following sections of tier-1 and/or tier-2 is for clarity purposes only. Farm Smart 2000's tier-3 is used for assessing the effectiveness of cropping plans based on the producer's farming conditions and previous cropping practices. The architecture of tier-3, and its integration with the remainder of Farm Smart 2000 is shown in Figure 5.3. The user interface is folder-based, like SmartRKS, allowing the user to enter data in a convenient and intuitive method. The input provided by the user comprises a "what-if" scenario that outlines decisions the user is contemplating for the up-coming growing season. During data entry, the user can execute agents in tier-2 in an "interactive mode" to assist with the input of valid data.

At any time, an analysis can be initiated with the appropriate user's data being accessed from the SmartRKS and placed in the blackboard database to be combined with internal system data and analyzed by rules and agents. A rule-based global expert (i.e. Global Control Expert) controls the analysis of the data, and with the help of agents, produces results which are also stored in the blackboard database. When the analysis is complete, an interface displays the results to the user. The user is able to change values and parameters within each "what-if" scenario and analyze as many scenarios as desired. Once users are content with the results of a particular scenario and intend to use these decisions as their crop plan, the scenario can be "committed" to the SmartRKS.

6.4.1 Using Farm Smart 2000 Tier-3

When Farm Smart 2000 is initiated, the users can select “User Information” from the “Settings” menu and personalize their system by registering their name, or farm name, and address, and customizing the look of the main window by inserting a picture of interest (e.g. of their farm). This personalization will be presented on the main screen when the system is next started, and thereafter, until changed or removed. Normally users then initiate tier-2, the Advanced Decision Support, by opening the “tool box” and inserting available stand-alone expert systems to use in their decision-making process. Users are not restricted to filling the tool box with only Farm Smart 2000 expert systems but rather can include any programs (i.e. tools) for their convenience.

Tier-3, Crop Planning, provides producers with the capability to plan their cropping strategy for a particular growing season, usually, but not necessarily the next growing season. A producer might wish to “play” with the current or last cropping plan to determine if alternate plans might have been more cost effective or profitable. The Crop Planning module enables the user to build scenarios that consider some or all the producer’s fields in combination with some or all of the crops being contemplated.

After the user selects a farm and year from a dialog box, fields and crops are displayed on a selection grid from which the user chooses fields and crops to be analyzed. Crops are listed across the top of the grid and fields are listed down the left side of the grid. If the farm chosen has no fields defined, the scenario building cannot continue; fields must be defined in the SmartRKS. One or more cells (i.e. the intersection of a row and column) can be selected on the grid. The intersection (i.e. the cell) contains one of the following messages:

- 1) Needs Evaluation - The crop and field combination have not been evaluated in which case the user clicks the evaluate button and an analysis is performed to determine the suitability of growing the selected crop on the selected field.
- 2) Evaluation is Current - The crop and field combination has been evaluated enabling the “Show Inputs” and “Show Results” buttons to be selected by the user to obtain further information about he selected crop and field.

- 3) Evaluation is Out of Date - Some of the inputs were changed since the scenario was last evaluated necessitating the evaluate button to be clicked to update the evaluation.

6.4.1.1 Buttons

The farm and year selected above are shown in the grid window along with buttons above and below the grid. Clicking these buttons performs a task or opens a dialog box for further input and include:

- Add Location...
- Add Crop...
- Modify Location... (displayed when a single location row is selected)
- Delete Location... (displayed when a single location row is selected)
- Modify Crop... (displayed when a single crop column is selected)
- Delete Crop... (displayed when a single crop column is selected)
- Show Inputs
- Evaluate
- Show Results
- Commit to Record Keeping System... (displayed when a maximum of one cell per row is selected)
- Print
- New Scenario...
- Open Scenario...
- Save Scenario...
- Save Scenario As...
- Exit Module

The functionality of these buttons is described in the following sections.

6.4.1.1.1 Add or Modify a Location

Location refers to the field or area where the crop will be planted. The same dialog box is used for each function, either entering data about a location or changing data about a location. More than one field and/or management unit can be selected for evaluation.

As fields (and management units, if applicable) are selected, data from the SmartRKS is automatically loaded into various boxes within the window (e.g. soil zone, soil texture, soil pH, percent soil organic matter). Users need to fill in blank boxes (e.g. depth of moist soil, seed bed moisture, seed bed soil temperature). The data fields include:

- Field - through the drop down list box, the user selects a field from the pick list for the location;
- Management Unit - through the drop down list box the user selects a management unit, if applicable, from the pick list. This box is inactive (i.e. gray) if no management units have been defined for the field selected.

Note: Once the field and management unit are selected, the previous crops grown are listed in the dialog box.

- Soil Zone - displays the soil zone of the selected field or management unit;
- Soil Texture - displays the soil texture of the selected field or management unit;
- Soil pH - displays the soil pH of the selected field or management unit;
- Soil Organic Matter - displays the percent soil organic matter of the selected field or management unit;
- Depth of Moist Soil - the user enters the depth of soil moisture expected at the time of seeding. The units can be changed using a drop down list box available;
- Seed Bed Moisture - using the drop down list box, the user selects the seed bed moisture (e.g. dry, normal, wet) expected at the time of seeding;
- Seed Bed Soil Temperature - using the drop down list box, the user selects the seed bed soil temperature (e.g. cool, normal, warm) expected at the time of seeding.

When all the required information has been entered, the user uses the “add” or “modify” button to store the location information into the scenario being created. If a location has been added, it will appear as a row heading on the selection grid. Clicking the “cancel” button returns the user to the selection grid without using the information.

6.4.1.1.2 Add or Modify a Crop

This dialog box enables the user to enter and change information about the crop(s) being considered for seeding this year. More than one crop can be selected for evaluation.

The data fields include:

- Crop to Grow in (year) - using the drop down list box, the user selects from the pick list the crop that is intended to be seeded;
- Variety - from a drop down list box, the user selects from the pick list the variety that is intended to be seeded;
- Seeder - using the drop down list box, the user selects a seeder they have available for seeding this crop. The list of seeders available in the pick list is derived from their equipment inventory contained in SmartRKS;
- Target Yield - the user enters the number of units (e.g. bushels) per land unit (e.g. acre) that the crop is expected to yield. The units displayed are the units selected in the preferences within SmartRKS but can be changed by clicking on the drop down list boxes and selecting the desired units;
- Seeding Depth - the user enters the depth at which the seed will be placed. The units displayed are the units selected in the preferences with SmartRKS but can be changed by clicking on the drop down list box and selecting the desired unit;
- Seeding Rate - the user enters the number of units (e.g. bushels) per land unit (e.g. acres) that they intend to seed. The units displayed are the units selected in the preferences within SmartRKS but can be changed by clicking on the drop down list boxes and selecting the desired units;
- Seeding Date - the user enters the intended seeding date for this crop in the format mm/dd/yyyy format, or clicks on the calendar button and selects a date;
- Cost of Seed - the user enters the cost of seed per unit displayed;
- Expected Sale Price - the user enters the price per unit displayed that they expect to receive for this crop.

When all the required information has been entered, the user uses the “add” or “modify” button to store the crop information into the scenario being created. If a crop has

been added, it will appear as a column heading on the selection grid. Clicking the “cancel” button returns the user to the selection grid without using the information.

6.4.1.1.3 Function of Other Buttons

The functionality of the other buttons residing around the selection grid are described below. The function of many of these buttons can also be accessed either via toolbar buttons or from the File menu.

- **Delete Location** - deletes the selected location from the scenario.
- **Delete Crop** - deletes the selected crop from the scenario.
- **Show Inputs** - this button enables the user to view the input data for the scenario/evaluation. See Section 6.4.1.5 for more details.
- **Evaluate** - the evaluate button instructs Farm Smart 2000 to assess the suitability of the selected crop(s) for the selected location(s). Once the evaluation is complete, the message at the intersection(s) of the selected crop(s) and location(s) changes to “Results are current”.
- **Show Results** - this button enables the user to view the results of the evaluation(s). See Section 6.4.1.6 for more details.
- **Commit to Record Keeping System** - this button enables the data entered in the “location” and “crop” dialog boxes to be saved in the SmartRKS. This button is activated only when a maximum of one cell per row is selected (e.g. only one crop and its related inputs per location can be committed at one time).

During the commit process, all the inputs for the selected crop/location combinations are examined and records are created in the SmartRKS based on the data provided. Before the commitment is actually made, a dialog box lists the changes that are about to be made and waits for the user’s authorization to proceed. If the user provides the authorization to proceed by clicking on the “yes” button, Farm Smart 2000 will update SmartRKS records as applicable including:

- update Location information
- add or update a Seed Input record
- create a Seeding Event record

- add Tillage Event records
- update Tillage Equipment records
- create Weed and Weed Control events
- create Disease and Disease Control events
- create Insect and Insect Control events
- create a Soil Test event
- create Fertilizing events
- update Fertilizer Input records

A data problem might arise later which is described in Section 6.4.1.2.

- **Print** - this button enables the user to make a hardcopy of results and is described further in Section 6.4.1.7.
- **New Scenario** - this button enables the user to create a new scenario. If the user is currently working with an existing scenario which has not been saved, they are given the opportunity, through a dialog box, to save it before continuing with a new scenario.
- **Open Scenario** - this button enables the user to open a previously-saved scenario from a list presented. The user is prompted to save the current scenario being work with, if applicable, before a new one is opened. In addition, if the fields on which the scenario is based have been deleted, the user will be informed and the scenario will not be opened.
- **Save Scenario** - the user clicks this button to save a new scenario or to save changes to an existing scenario. A previously-saved scenario will be saved with its original name. If a different name is desired, then the user clicks on the “Save Scenario As...” button.
- **Save Scenario As** - enables the user to specify a new name for an existing scenario.
- **Exit Module** - this button closes the Crop Planning module (i.e. tier-3) of Farm Smart 2000 and returns the user to the main screen. If necessary, the user is given the opportunity to close an open scenario before exiting.

6.4.1.2 Work-Ahead Problem

As mentioned, the entire process of handling data in a multi-agent cooperative problem-solving system can be difficult and with the tightly integrated data within Farm Smart 2000, data handling is complex. One such data handling problem, called the work-ahead problem, is difficult to cope with throughout Farm Smart 2000 and is introduced below with more problem detail being described in Section 6.4.3.6.

In Farm Smart 2000's databases, there are predefined pest, pesticide, and crop variety data which change from year to year and are updated with each new release of Farm Smart 2000. The work-ahead problem arises if the user has created records in SmartRKS or scenarios in tier-3 for a particular year prior to receiving the updated data. When these records are accessed, it is possible that data that was valid in a previous year, is now no longer valid once the predefined data is updated. For example, the user may find that a crop entered last year and intended to be grown this year, has been de-registered, or maybe the same is true for a herbicide that was planned for use.

To handle this problem, when Farm Smart 2000 is first started with the new updated data, it verifies that all the SmartRKS records are valid. If any affected records are detected, the user is prompted to correct these records by selecting one of the following options:

- 1) Keep this variety by changing the year acquired to <latest year in which variety existed>;
- 2) Choose an alternate variety/fungicide/herbicide/insecticide from <drop list containing choices>;
- 3) Delete the record and all the events that access it.

Furthermore, when the system first loads a scenario with the updated data in place, it checks for this situation and if affected records are detected, the user is prompted to:

- 1) Choose an alternate variety/fungicide/herbicide/insecticide from <drop list containing choices>;
- 2) Delete the affected part of the scenario.

Once the affected records are corrected, the user does not see these messages again.

6.4.1.3 Using the Selection Grid

As introduced above, the Selection Grid enables the user to choose which crops and locations are to be analyzed and reported on. The grid is initially empty and the user can use the following procedure to create the locations and enter the crops to be evaluated.

- 1) Use the “Add Location” button to select as many locations as desired to evaluate;
- 2) Use the “Add Crop” button to select the crops that are being considered to grow on these locations;
- 3) Select the entire grid by clicking the cell box in the upper left hand corner which then changes the color of the grid indicating that it has all been selected;
- 4) Click the “Show Inputs” button and enter inputs that will apply to all location/crop combinations.
- 5) Return to the Selection Grid, and select a single row (i.e. one location with one or more crops);
- 6) Use the “Show Inputs” button and enter the inputs that will apply to that location, and all other locations on the grid, by repeating steps 5 and 6;
- 7) Return to the Selection Grid, and select a single column (i.e. one crop with one or more locations);
- 8) Use the “Show Inputs” button and enter the inputs that will apply to that crop, and all other crops on the grid, by repeating steps 7 and 8;
- 9) Return to the Selection Grid, and if applicable, select a single cell or a small group of cells;
- 10) Use the “Show Inputs” button and enter inputs that will apply specifically to growing that particular crop at that location;
- 11) Save the scenario;
- 12) Select some or all of the grid for evaluation and use the “Evaluate” button;
- 13) To review the results, click the “Show Results” button;
- 14) The routine of “Edit Inputs”-“Save Scenario”-“Evaluate”-“Show Results” is performed until the user is satisfied with the results;
- 15) Generate/print any reports desired;

- 16) Commit data to SmartRKS by selecting one cell from each row (i.e. one crop for one location) representing the crop planning decision, and then click the “Commit to RKS” button. Any missing data can be entered via SmartRKS.

6.4.1.4 Scenarios

A scenario is a set of data that is used for evaluating the effectiveness of planned crops on particular parcels of land. A new scenario is created whenever data is changed and can be evaluated and saved for future reference. Changes to scenarios can be saved by selecting “Save Scenario” and new scenarios can be saved by selecting the “Save Scenario As ...”, both from the “File” menu. Scenarios are opened by selecting “Open Scenario...” from the “File” menu. All these functions are also available as toolbar buttons. If the fields on which the scenario is based have been deleted, the scenario cannot be opened.


6.4.1.5 Input Folders

The input folders window has two rows of tabs as shown in Figure 6.5. Details of farm, year, crop and location are listed above the tabs for easy reference. Clicking on a tab opens a folder window below the row of tabs from which the user can view, select, or enter data pertaining to their farming operations. Data previously entered in SmartRKS is accessible for use in calculating results.



Figure 6.5: Tabs in the Input Folders Window

Crop Planning information for each crop and location combination previously selected on the Selection Grid, is usually entered by initially selecting the conservation goal tab and then progressing through the remainder of the tabs, with their respective folders, entering required data. Data that has been entered in the folder windows can be modified if required by clicking the Event item to be modified, making the required changes to the data entered previously, and selecting the “Modify” button to save the changes. In the same way, an Event item can be removed from a scenario by selecting the event item to be removed and clicking the Delete button and clicking Yes to confirm the deletion. The tabs and their respective folder windows are introduced in the following sections. The user can

return to the Selection Grid at any time by clicking the Selection Grid  toolbar button or by selecting Selection Grid from the Scenario menu.

6.4.1.5.1 Goals Folder

The Goals tab opens the Goals Folder window and presents a drop down list box from which the user selects their conservation goal. Once selected, this goal is used during the evaluation of the user's crop plans.

6.4.1.5.2 Tillage Folder

The Tillage Folder window is used for entering information about tillage operations, if any, that the user intends to perform on this location this year. More than one tillage operation can be used in the evaluation. Other information to enter in the Tillage window includes:

- 1) Implement Name - from a drop down list box containing implements the user previously entered into SmartRKS, the user selects the implement to be used for the tillage operation.
- 2) Type of Implement - is displayed and can only be changed by accessing SmartRKS.
- 3) Number of Passes - that the user intends to make during this tillage operation.
- 4) Timing of Passes - from a drop down list box, the user selects the time at which they intend to perform this tillage operation (e.g. spring, pre-seed).
- 5) Cost per Pass - is the cost per land unit for this tillage operation (e.g. \$1.25/acre). The units displayed (e.g. acres) are the units that the user selected in the SmartRKS preferences, although these units can be changed by selecting the appropriate units from a drop down list box.

When all information has been entered the "Add" button places this information into the scenario the user is building. The user can use the "Modify" button to alter an existing tillage operation whose information has changed, the "Delete" button to remove the selected tillage operation from the scenario, or the "Cancel" button to cancel any changes without saving them. Several tillage operations can be added to the scenario.

6.4.1.5.3 Weed Control Folder

The Weed Control Folder window enables the user to enter information about potential weeds on this location this year. More than one weed combination can be submitted for evaluation. However, the Work Ahead Problem described in Section 6.4.1.2 might affect future scenarios. The Weed Control Folder window contains the following information:

- 1) Weed 1 - from a drop down list box containing an extensive list of weeds, the user selects the name of the weed that they might encounter on this location in this year. In the associated “Density” box, the user selects from a drop down list, the expected density of this weed. If applicable, additional weeds (e.g. Weeds 2, 3, and 4) are selected with their expected densities.
- 2) Herbicide Applied - the user may obtain help in selecting a herbicide or herbicide combination to be applied to this weed combination by clicking on “Calculate”. Internally, the Global Control Expert initiates (i.e. calls upon) the Crop Protection agent to provide alternative herbicides, rates, and costs to combat these weeds, which are then entered into the Weed Control Folder window upon completion of the calculation. If any information from the calculation is incorrect, it can be revised accordingly.

The “Add” button is used by the user to add this information to the scenario being created. Similarly, the “Modify” button is used to alter existing weed and herbicide information, the “Delete” button to remove selected weed and herbicide combinations from the scenario, and the “Cancel” button to cancel changes without saving them.

6.4.1.5.4 Disease Control Folder

The Disease Control Folder window enables the user to enter information about potential diseases in this crop this year. More than one disease combination can be submitted for evaluation. Again, the Work Ahead Problem described in Section 6.4.1.2 might affect future scenarios. The Disease Control Folder window contains the following information:

- 1) Disease - from a drop down list box containing a list of diseases, the user selects the name of the disease that they might encounter in this crop in this

year. In the associated “Severity” box, the user selects from a drop down list, the expected severity of this disease (e.g. low, moderate, or high).

- 2) Fungicide Applied - the user may obtain help in selecting a fungicide to apply to this disease by clicking on the “Calculate” button. In a similar manner as previously mentioned, the Global Control Expert initiates the Crop Protection agent to provide alternative fungicide options, rates, and costs to combat the disease(s), which are then entered into the Disease Control Folder window upon completion of the calculation.

The user adds this information to the scenario being created by clicking on the “Add” button. The “Modify” button is used to alter existing disease and fungicide combinations, the “Delete” button to remove selected disease and fungicide combinations from the scenario, and the “Cancel” button to cancel changes without saving them.

6.4.1.5.5 Insect Control Folder

The Insect Control Folder window enables the user to enter information about potential insects in this crop or field this year. More than one insect can be submitted for evaluation. The Work Ahead Problem described in Section 6.4.1.2 might affect future scenarios. The Insect Control window contains the following information:

- 1) Insect - from a drop down list box containing a list of insects, the user selects the name of the insect that they might encounter in this crop or field in this year. In the associated “Infestation Level” box, the user selects from a drop down list, the expected severity of the infestation of this insect (e.g. low, moderate, or high).
- 2) Insecticide Applied - the user may obtain help in selecting an insecticide to combat this insect by clicking on the “Calculate” button. Again, the Global Control Expert initiates the Crop Protection agent to provide alternative insecticide options, rates, and costs to eradicate the insects, which are then entered into the Insect Control Folder window upon completion of the calculation.

The user adds this information to the scenario being created by clicking on the “Add” button. The “Modify” button is used to alter existing insect and insecticide

combinations whose information has changed, the “Delete” button to remove selected insect and insecticide combinations from the scenario, and the “Cancel” button to cancel changes without saving them.

6.4.1.5.6 Fertility Folder

The Fertility Folder window is used for entering and viewing soil test and fertilizer data. Soil test nutrient levels and required soil nutrients are normally available from soil test laboratory reports. In the Fertility Folder window, the user selects the units of quantity of nutrient per land unit (e.g. kg/ac, lb/ac, lb/ha) via drop down list boxes. If the nutrient data from the soil test results have been entered in SmartRKS, it will automatically appear in the Fertility window in the “Required Levels” and “Desired Levels” boxes. Else, clicking on the “Default” button will complete the “Desired Levels” (of nutrients) boxes with the “Required Levels” data to facilitate the calculation of nutrients still required. The Desired Levels can be modified by the user if they want to treat this field or management unit differently. Within the Fertility window, using the “Amounts Applied” data (from Events in SmartRKS), a calculation is performed to show the amount of nutrients still required. In the fertilizer part of the window, the user enters a fertilizer and application rate that will satisfy the figures shown in “Amounts still Required”. The user also selects equipment to be used for fertilizer application from the drop down list box. The implements available in the drop down list box are those that were entered in “Equipment” in SmartRKS. In association with the fertilizer specified, the user enters the formulation as 4 numbers (e.g. 16 20 0 0) for percentages of Nitrogen, Phosphorus, Potassium, and Sulfur. Other data to be entered by the user in this portion of the Fertility Folder window include:

- Nitrogen Source - which is selected from a drop down list box (e.g. Ammonium Nitrate, Anhydrous Ammonia, manure, or urea);
- Application Method - which is selected from a drop down list box (e.g. broadcast, deep banded, injected, nested, seed placed, or side banded);
- Application Rate - which is the intended rate to be used in the units selected at the top of the Fertility Folder window;
- Fertilizer Cost - which is the cost of the fertilizer per unit selected.

An area at the bottom of the Fertility Folder window shows the user, for historical reference, other fertilizer applications on this location. If the user is content with the nutrient amounts for this fertilizer application, the “Add” button can be selected. Otherwise, if the “Amounts Still Required” are not satisfactory, the user can change the application rate or formulation as required and select the “Modify” button to produce a new set of “Amounts Still Required”. The user can continue fine tuning in this manner as required. If the “Delete” button is clicked, the selected fertilizer application is removed from the scenario. The “Cancel” button is used to cancel changes without saving them. As many fertilizer applications as desired can be added.

6.4.1.5.7 Weather Folder

The Weather Folder window enables the user to enter information about expected precipitation and temperature for this year. Only one precipitation-temperature combination can be selected per evaluation. These values are selected in drop down list boxes (e.g. below average, normal, or above average). This weather information is used in the assessment of all possible location-crop scenarios.



6.4.1.5.8 Economics Folder

The Economics Folder window is used for entering information regarding maximum costs for inputs. For example, from experiences in past years, the user might have determined that it is unprofitable to expend more than \$30.00 per acre for fertilizer. Farm Smart 2000 uses this economic information to advise the user if a planned course of action exceeds the maximum input costs set. If desired, the user can leave any or all of these items blank.

The user selects an economic unit via a drop down list box which applies to all the maximum costs (e.g. acre or hectare). The maximum cost values that can be entered include:

- Maximum cost of seed;
- Maximum herbicide cost;
- Maximum fungicide cost;
- Maximum insecticide cost;
- Maximum cost of fertilizer;

- Maximum cost of tillage operations.

At this stage the user can evaluate the data entered and view the results of the scenario created. To evaluate the data, the user clicks the Evaluate  toolbar button or selects Evaluate from the Scenario menu. Otherwise the user can at any time, click on another folder tab or return to the Selection Grid by clicking the Selection Grid  toolbar button or by selecting Selection Grid from the Scenario menu.

6.4.1.6 Result Folders

To view the results of an evaluation, the user can click the “Results” tab from the Inputs/Results window, or click on the “Show Results” button on the Selection Grid. If Results cannot be selected from the menu or toolbar because it is grayed out, there is insufficient data available in the Crop Folder or the Soils and Location Folder as there must be at least one location contained in the Soils and Location Folder, and at least one crop in the Crop folder in order for “Results” to be available. In the input/results window, a second set of tabs will appear on the right side of the screen as shown in Figure 6.6, once results are available.

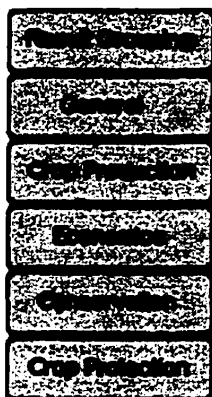


Figure 6.6: Result Tabs

6.4.1.6.1 Result Grouping

This Tab window contains two “Group By” boxes. The item which the user selects in the first “Group By” box will act as a heading in the reports. If applicable, the user can select another item in the “Then Group By” box which will act as a sub-heading in the reports. With these grouping options, the user can customize the layout and order of information in the report.

6.4.1.6.2 General Results Folder

The General Results Folder window is comprised of a report based on the Crop and Location combination the user selected and appear according to the selections in the “Group By” boxes. The report contains the following information:

- Crop year
- Farm
- Crop
- Location
- Conservation Goal
- Erosion Risk
- Herbicide Residue Risk
- Potential Yield
- Gross Return
- Total Expenses
- Net Return

Some calculated results are presented in the color blue, which, if the user clicks on, will provide a brief description of related information.

6.4.1.6.3 Crop Production Results Folder

The Crop Protection Results Folder window is a report, based on the Crop and Location combination the user selected and is presented according to the selections in the “Group By” boxes. The report contains information related to the following:

- Crop
- Location
- List of previous crops
- Herbicide Residue Risks
- Seeding Date
- Harvest Date (i.e. estimated)
- Seeding Depth
- Seed Bed Temperature
- Target Yield, Potential Yield, Variety Average, Break-Even Yield

- Fertilizer Summary (including levels required, desired, applied, and still required for Nitrogen, Phosphorus, Potassium, and Sulfur)

Some calculated results are presented in the color blue, which, if the user clicks on, will provide a brief description of related information.

Within this same window, there are a “Fertilizer Application Details” button and a “Yield Graph” button. If selected, the “Fertilizer Application Details” button will produce an additional window in which specific details about fertilizers applied and their placement are reported. Selecting the “Yield Graph” button will produce a graph of the Target Yield, Potential Yield, and Variety Average Yield, and Break-Even Yield for each Location and Crop combination that was chosen on the Selection Grid. By clicking the arrow beside the Location/Crop drop down list box, a particular Location and Crop combination can be selected to be graphed.

6.4.1.6.4 Economic Results Folder

The Economic Results Folder window is based on the Crop and Location combinations selected by the user and are displayed according to the selections made in the “Group By” boxes. The report contains the following economic information, based on dollars/acre or dollars/hectare:

- Gross Return
- Seed Costs
- Tillage Costs
- Herbicide Costs
- Fungicide Costs
- Insecticide Costs
- Fertilizer Costs
- Total Costs
- Net Return

For each item of information, three sets of results are presented:

- Calculated Results - these are calculated from the data the user entered.
- Typical Results - these results represent costs on farms similar to the users.

- **Maximum Results** - these results are based on the maximum costs the user entered in the Economics Folder window, if any. Some calculated results are presented in the color blue, which, if the user clicks on, will provide a brief description of related information.

Three buttons are available in the Economics window, the “Cost Comparison Graph” button, the “Cost Breakdown” button and the “Cost Details” button. The “Cost Comparison Graph” button provides a bar graph of Maximum, Typical, and Actual costs for all cost items for each Location and Crop combination that was chosen on the Selection Grid. The cost items include:

- Seed
- Tillage
- Herbicide
- Fungicide
- Insecticide
- Fertilizer

By clicking the arrow beside the Location/Crop drop down list box, a particular Location and Crop combination can be selected to be graphed. The “Cost Breakdown” button produces a pie graph of all the same Calculated Cost items listed above for all Location and Crop combinations that were chosen on the Selection Grid. By clicking the arrow beside the Location/Crop drop down list box, a particular Location and Crop combination can be selected to be graphed. The “Cost Details” button displays a list of all Calculated Costs (see list above) for all Crop and Location combinations that were chosen on the Selection Grid.

6.4.1.6.5 Conservation Results Folder

The Conservation Results Folder window is a report based on the Crop and Location combinations the user selected, displayed according to the user’s choices in the “Group By” boxes, and contains the following conservation information:

- Soil Zone and Soil Texture
- Soil Organic Matter
- Conservation Goal

- Seeder Used
- Harvest Last Fall (i.e. Combine, Crop, Yield, Cutting Height, Residue Level After Harvest)
- Total Residue Loss
- Residue Level Before Seeding
- Erosion Risk This Spring
- Erosion Risk Next Spring

Some calculated results are presented in the color blue, which, if clicked on, will provide a brief description of related information. The “Tillage Details” button available in this window provides a list of all tillage operation details (i.e. Implement, Timing, Number of Passes, Residue Loss/Pass) for all Crop and Location combinations that were indicated on the Selection Grid.

6.4.1.6.6 Crop Protection Results Folder

The Crop Protection Results Folder window is a report based on the Crop and Location combinations the user selected, displayed according to the user’s choices in the “Group By” boxes, and contains the following crop protection information:

- Future Cropping Restrictions
- Potential Weed Resistance Problems
- Potential Disease Problems
- Potential Yield Loss from Disease

Some calculated results are presented in the color blue, which, if the user clicks on, will provide a brief description of related information. If a result is marked <Unknown>, there is insufficient data with which Farm Smart 2000 can base a calculation.


The Crop Protection Result Folder window contains several buttons including the following:

- Weed Identification button - provides comprehensive information about the weeds the user expects to encounter.
- Herbicide Details button - displays a list of all herbicide applications for all Crop and Location combinations that were chosen on the Selection Grid and

indicates any weed resistance problems and inaccuracies in herbicide selections that might be present.

- **Fungicide Details button** - lists all fungicide applications, costs, and application rates versus recommended rates for all Crop and Location combinations that were chosen on the Selection Grid.
- **Insecticide Details button** - provides a list of all insecticide applications including inaccurate insecticide selections for all Crop and Location combinations that were chosen on the Selection Grid.

6.4.1.7 Printing Reports

Results are printed by clicking the Print... button on the Selection Grid, by selecting Print on the File menu, or by clicking the Print  button on the Inputs/Results page toolbar. There must be at least one cell selected on the Selection Grid for these print commands to be active. Prior to printing reports the user can check and/or change the default margins by selecting Page Setup on the File menu.

The print dialog box which appears after selecting Print, allows the user to choose the particular reports desired. The user clicks the boxes beside the “Inputs” and “Results” titles to select the report(s) (i.e. folders) to be printed. The “Select All” and “Deselect All” buttons are used to select or clear all applicable boxes. The names of the Result Folders cannot be selected unless all the selected cells on the Selection Grid have been evaluated. If any of the cells selected contain out-of-date evaluation information, the user has the option to cancel the print operation. If the user proceeds, a note to this effect appears in the report. The groupings selected in the Result Grouping tab window apply to the result reports.

6.4.2 Farm Smart 2000 Tier-3 System Components

This section describes the system components in Farm Smart 2000 tier-3 as illustrated in Figure 6.7. In Figure 6.7, an arrow from component A to component B indicates that component A provides component B with some type of resource (e.g. data) or facility (e.g. analysis, calculation). The four main agents, the Weed Management Planner, Crop Protection Planner, Crop Variety Select, and Crop Rotation Planner, as previously described, are integrated into tier-3.

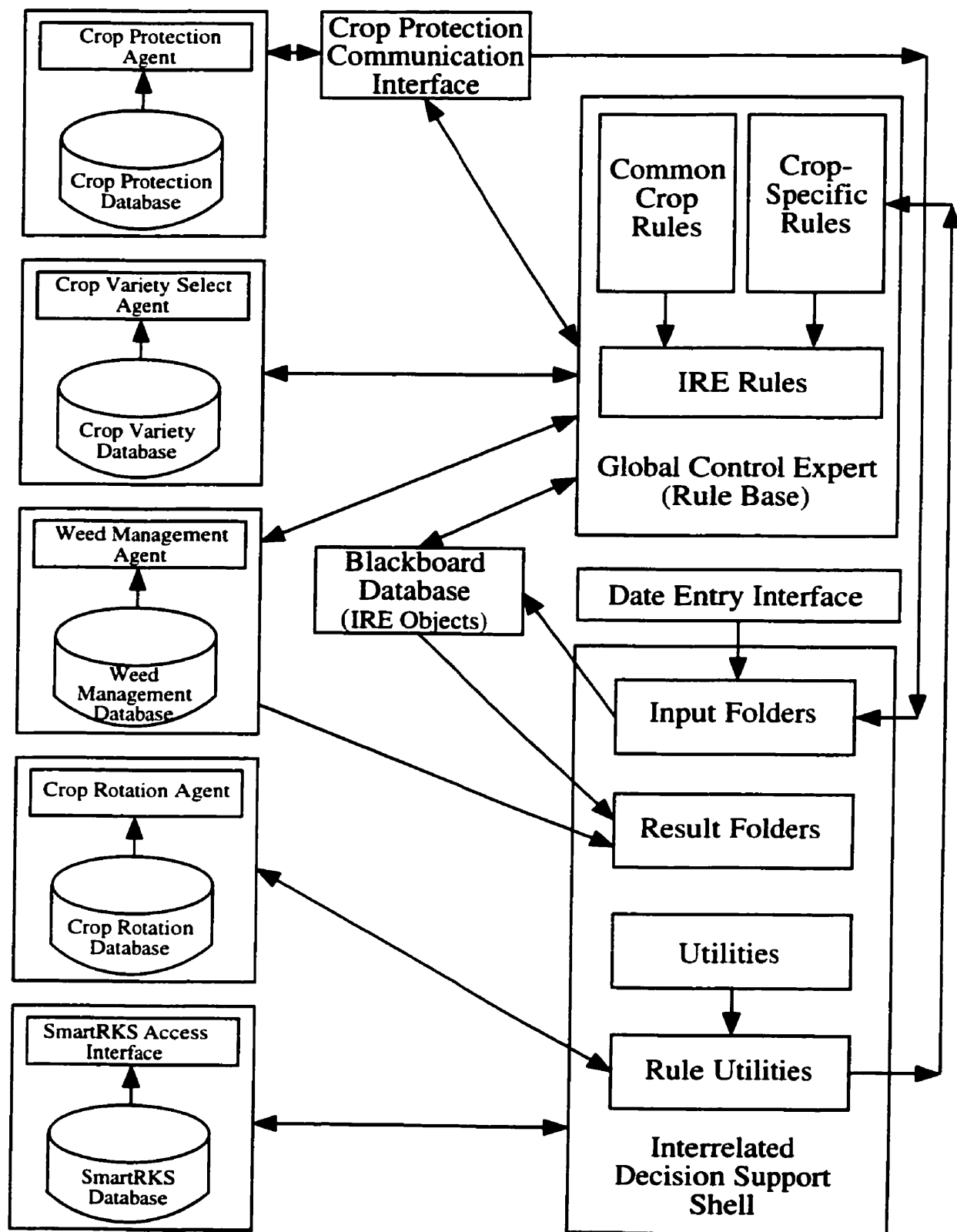


Figure 6.7: Tier-3 System Components

The majority of tier-3's operational ability resides in the Interrelated Decision Support Shell module which interacts with the user via the Input and Result Folders. Neuron Data's *Open Interface Element*[™] (i.e. a software toolbox of callable routines) is used to implement the user interface in the Interrelated Decision Support Shell (See Figure 6.7). Neuron Data's *Intelligent Rules Element*[™] (IRE) is used to implement the evaluation process within the Global Control Expert (See Figure 6.7). The Global Control Expert (i.e. IRE) consists of rules and an inference engine to process the rules (See Figure 6.7). Data is exchanged between the Interrelated Decision Support Shell and the IRE Rules component via the IRE objects, or Blackboard Database.

6.4.2.1 Description of Components

This section provides a description of the components in Figure 6.7, Tier-3 System Components, other than the agents which have been described previously.

- 1) Interrelated Decision Support Shell - contains the user interface and the process logic to control tier-3. The Shell consists of several sub-components which include:
 - a) Input Folders - implement the user interface and through underlying objects (i.e. Object Oriented Design), enable data entry.
 - b) Result Folders - implement the user interface and through underlying objects, generate and display the results of an evaluation.
 - c) Rule Utilities - implement "Callback" functions that the IRE rule inference engine can call. These functions provide more sophisticated processing than can be performed with the IRE rule language. Some examples include:
 - Interact with the Crop Rotation Agent
 - Retrieve data from the SmartRKS Database
 - Implement the yield model which calculates the possible yield result and supporting agronomic reasoning information.
 - d) Utilities - consist of a number of internal utility functions that provide processing that is common to both the Interrelated Decision Support Shell and Rule Utilities. Some examples include:


- Implement Dynamic Data Exchange (DDE) with various agents
 - Exchange data with the IRE objects
- 2) **Rule Base** - contains all of the rules required to process an evaluation. The rules are implemented using rules created within Neuron Data's IRE. It contains two sub-components which include:
 - a) **Common Crop Rules** - implements rules that are common to all crops
 - b) **Crop-Specific Rules** - implements rules that are specific to particular crops.
 - 3) **Blackboard Database** - contains the current set of known "facts" (i.e. at the time the rule engine is executing) and transfers data, via IRE objects, between the Input Folders and the Rule Base (i.e. IRE Rules) and between the Rule Base (i.e. IRE Rules) and the Result Folders. IRE Rules require its data to be stored in the IRE's own object model. Thus, it is necessary to transfer data from the Input Folders into IRE objects for processing, and then transfer the results back into the Result Folders.
 - 4) **SmartRKS Access Interface** - is an API that provides a single point of access to the SmartRKS database. The SmartRKS database contains the user's previous farm characteristics and operations as well as internal system data such as predefined lists of supported crops, varieties, pests, and pesticides.
 - 5) **Crop Protection Communication Interface** - is an utility in the form of a DLL that provides communication (i.e. transfers data) with the Crop Protection Agent
 - 6) **Date Entry Interface** - is a utility in the form of a DLL that provides a pop-up calendar for use when entering dates in the Input Folders.

6.4.2.2 Rules

In the Interrelated Decision Support Shell, a rule is a process that includes mapping one or more input values, either from the Input Folders or from the SmartRKS database, to one or more results that are then displayed in the result folders. Some rules are implemented in the IRE, using its internal rule language, whereas more complex rules require the use of other resources and facilities. The IRE has a "callback" function which

enables its rule engine to call external utilities such as an agent or a callback-utility to provide the desired results.

6.4.2.3 The Analysis/Evaluation Process

The analysis/evaluation process performed in the Interrelated Decision Support component is illustrated in Figure 6.8. In Figure 6.8, the arrows indicate the flow of control through the process. A  symbol means iteration takes place where the operation is carried out multiple times. Items enclosed in square brackets (i.e. “[]”) are called “conditions” and only allow the associated arrow (i.e. path) to be followed if the condition is true. When an arrow begins at a thick horizontal line, called a “synchronization bar”, the operations being pointed at by the arrows can be executed in any order or in parallel. When arrows end at a synchronization bar, the sequence of operations that end at the bar must be completed before proceeding past the bar. If a single arrow terminates at a synchronization bar, there is an iteration that has been started previously (i.e. further up the path) which must be completed before proceeding.

6.4.3 General User Requirements/Specifications

Farm Smart 2000 tier-3 was designed to meet the following requirements:

- Accept data entry from the user regarding proposed farming operations for the upcoming growing season based on a single location (i.e. field or management unit), or several locations, and in combination with a single crop or several crops;
- Evaluate all the data available for each crop/location combination generating as many results as possible, which consist primarily of quantitative values with supporting agronomic reasoning;
- Present the results in a clear, comprehensible format, summarizing information where possible, but allowing the user to see the underlying details as desired including the agronomic reasoning for all recommendations and results;
- Save different crop planning scenarios and commit the data from a final scenario to the SmartRKS record keeping system;

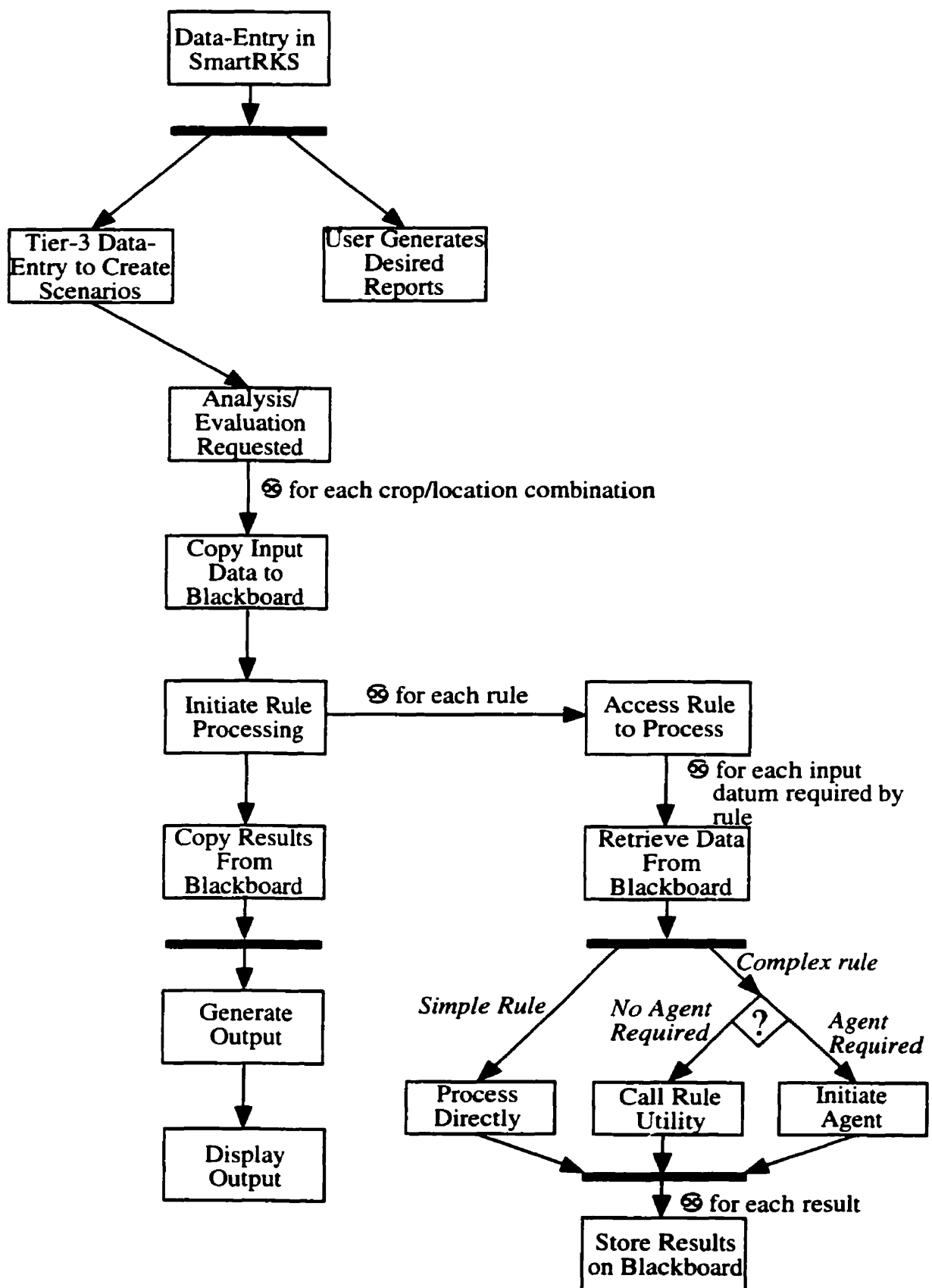


Figure 6.8: The Analysis/Evaluation Process

- Use general and crop-specific rules when processing evaluations;
- Utilize agents to perform the evaluations and analyses;
- Detect missing data and inform the user as to what data is required in order to calculate the result.

In fulfilling these requirements and specifications, many problems arose, for most of which, solutions were developed, while some required compromises. The following sections describe and discuss some of these problems, solutions, and compromises.

6.4.3.1 Analyzing Multiple Crop and Location Combinations

Farm Smart 2000 tier-3, in an early prototype, was developed to support a single crop and a single location. In order to simplify system expansion, this design was maintained in a rule base. The processing of multiple crops and/or locations is implemented in C++ support and interface code. Initially, multiple crops and locations were supported by making the crop and, soils and location folders, resemble the other multiple event folders, in terms of appearance and functionality. Hence, the user could specify multiple copies of the crop, or soils and location folder, in a similar method as they can specify multiple events in the other folders, and the system would maintain these separate copies.

In processing an evaluation, the system looped through each crop/location combination using a control algorithm. During each cycle through the loop the system would pass all the data for that combination (and all of the other input folders) to the rule base, process the evaluation, retrieve the evaluation results, and store them for future use. In presenting the results to the user, a grid was displayed. Each column of the grid corresponded to a single location, while each row corresponded to a single crop. Thus, each cell corresponded to the results for a crop/location combination. Any combination of cells in the grid were user selectable, along with some grouping options, and the results for those cells could be viewed in a collection of result folders.

This was a reasonable first design at supporting multiple crops and locations. However, it had a number of limitations including:

- Information in other folders was dependent on the particular crop and location chosen. For example, different locations would have different weed

problems, and different herbicides must be used on different crops. Thus, the chosen herbicide applications should depend on both the crop and location selected. The original design had no method to represent this dependency;

- Users were confused by the similarity of the crop, and soils and location folders, to the other multiple event folders. This was a result of the system considering all of the multiple events in each evaluation, but only one crop and one location in each evaluation;
- The results were not combined or summarized in any particular manner.

These limitations led to the development of a new approach. To make data entry more flexible, it was desirable to allow data to be edited in more than one crop/location combination (called a "SnapshotSet") at one time. Unfortunately, the input folders were capable of displaying only a single SnapshotSet at one time. To overcome this limitation, a new class, EditSet, was developed. An EditSet and a SnapshotSet are similar in that they contain the same type of data. The major difference is that a SnapshotSet represents a single crop/location combination, while an EditSet represents one or more crop/location combinations. The data from all of the SnapshotSets that the user wishes to edit are "merged" to create an EditSet object.

This merge process is similar to the method used by toolbar buttons in a word processor. When text is selected in a word processor, the state of the toolbar buttons is determined by "merging" the state of the text formatting in the selection. Hence, for example, if all of the text in a selection is bold, the bold toolbar button is displayed in the down position. If the state of the text in the selection is not bold (i.e. plain), the bold toolbar button is displayed in the up position. If the state of the text in the selection contains both bold and plain text, the toolbar button is displayed in an "indeterminate" state. In this latter case, clicking on the bold toolbar button, changes all the selected text to bold. Clicking the bold toolbar button again, changes all the selected text to plain.

In a similar manner, if all of the SnapshotSets have the same value for a particular data item, that value will be displayed in the input folders. However, if any of the values differ in the SnapshotSets, blanks will be displayed in the input folders. If the user modifies any data value on the screen, that value will be changed in all of the associated

original SnapshotSets. If a blank value is left blank, then the original SnapshotSets will retain their original values.

It was also necessary to merge events, which was accomplished by assigning a unique identifier to each event. During the merging process, only events with the same identifier can be merged when merging individual events as described above. However, each event has at least one required data value. If the required data values are blank due to differences in the events, then the merge operation fails. The resulting EditSet contains the events that were successfully merged. All other events in the original SnapshotSets remain unchanged.

6.4.3.2 Results Presentation

With the merging of the inputs, the results generated by the rules are stored in a separate parallel hierarchy of result snapshots rather than directly within the input snapshots. However, the result snapshots, and their corresponding input snapshots, are contained in the SnapshotSet class.

The resultant “text” is generated by several OutputGenerator classes which produce a string that can be parsed by the *Elements Environment*[™] hypertext editor. Each generator can display several results with associated justifications as well as other text. Results having justifications display as hypertext links, such that when the user clicks on them, the justifications appear in a pop-up window.

The generators are represented and controlled by a number of OutputPage classes. Charts and graphs are displayed under the control of other OutputPage classes. The generators classed as global, generate output that does not depend on any particular SnapshotSet.

6.4.3.3 Missing Data

The first prototype of Farm Smart 2000’s rule base used backward chaining as a primary inference mechanism, where it began by attempting to confirm a single hypothesis and work backward until it reached input data or could not proceed any further. This approach was problematic as it did not handle missing data very well.

To correct this problem, the rule base was redesigned to function in a data-driven or forward-chaining method. All of the user-entered input data is passed to the rule base,

which infers all of the results it can, given that data. To accomplish this, any inputs, not specified by the user, are set to the *Elements Environment*TM special value NOTKNOWN, which means that the value is explicitly not known, whereas UNKNOWN means not yet known. In addition, some rules and utilities were modified to cope with the possibility that one or more of their inputs might be NOTKNOWN.

There is currently an error (i.e. bug) in Neuron Data's *Elements Environment*TM (to be corrected in the next major maintenance release) that prohibits a value to be explicitly set to NOTKNOWN from C++. To manage this error, all input values are given a default value of NOTKNOWN in *Elements Environment*TM and then any unspecified user values are not passed to the rules resulting in the default NOTKNOWN value being used.

6.4.3.4 Managing Management Units

The original version of Farm Smart 2000 dealt with data at the field level. To support management unit data, several issues needed to be resolved including:

- different management units in the same field could have different soils and location information, as well as different historical events in the record keeping system;
- the user will probably want to manage different management units in different methods which impacts data entry in the input folders of Interrelated Decision Support.

To resolve these issues, management units were handled as fields. Hence, a location could be a field that has no management units or a management unit. Thus, when a field is user-selected that has management units, one of its management units must also be selected.

6.4.3.5 Delivery of User Explanations of Agronomic Reasoning

The results produced by Farm Smart 2000 are supported and justified by agronomic reasoning. Farm Smart 2000's reasoning capability explains the results to the user and thereby establishes and/or maintains credibility. Hence, it was desirable that the user be able to access this reasoning at any time without disadvantaging the presentation of results.

Utilizing the *Elements Environment*TM hypertext control was the best approach to present the results. The results appear as text in the window with agronomic explanations

appearing as hypertext links. Clicking on a hypertext link displays the agronomic explanation in a pop-up window. This approach provides flexible and extensible development opportunities and is easy for the users to understand.

6.4.3.6 Data Integration

One objective within the development of Farm Smart 2000, was to minimize data entry for the user, and further, to limit the necessity for the user to enter each element of data only once. One of the most difficult problems in the development of Farm Smart 2000 was implementing data integration. The following discussion will describe an example of one data integration problem which will serve to exemplify the degree of difficulty in solving data integration problems throughout Farm Smart 2000.

6.4.3.6.1 Problem Background

Tier-3, the Interrelated Decision Support component or crop planning component, and SmartRKS, the record keeping system, necessitate that much of the user's input data be selected from pre-defined lists in the system data, where available, such as crops and their varieties, pests, and pesticides. This stipulation ensures that the user cannot enter the same data with minor variations in spelling or naming convention. It also allows the crop planning component to know with certainty which entry the user is referring to. This certainty is what gives the crop planning component the ability to reason, based on the user's data in the SmartRKS record keeping system.

In order to support this input data model, it is necessary that the system data be maintained and frequently updated in order for Farm Smart 2000 to remain current and relevant. Some of the system data, such as crop varieties and pesticides in particular, undergo frequent changes from year to year as new varieties and products are registered and others are de-registered.

It is desirable that the users be able to select appropriate choices for the year in which they are currently entering data. For example, if users are entering 1995 data, they should not be presented with choices of pesticide products that were not introduced until 1998. Alternatively, they should be provided with choices of pesticide products that were not de-registered until after 1995.

To support this input data model, the system data is “versioned”. Each element of system data, or system object, is given a unique identification that is valid for the entire life of that data. The identification is used to uniquely refer to that object. Each system object is contained in one or more database records, referenced by the unique identification and a “validity range”, consisting of the first and last years for which that database record is applicable. Often, some supporting data for the system object (e.g. weeds eradicated by a herbicide) will change from year to year without changing the identity of the object itself. In these cases, several database records, with non-overlapping validity ranges, will exist to describe the object. Using this mechanism, an identification and the year in which the user is working can be utilized to locate the element of system data, and all supporting data, that the user is currently working with.

In order to enable users to enter data in years where system data is not available, the validity ranges were extended on the earliest system data records. For example, the earliest variety system data available in Farm Smart 2000 is 1991. Hence, if the user is entering variety data for a year prior to 1991, the 1991 variety data is used for years prior to 1991. Although this is not entirely accurate, it provides the capability to enter historical data²⁰. Analogously, the year ranges of the latest system data records were extended to allow for entry of future data.

6.4.3.6.2 Problem Scenario

The input data model described in Section 6.4.3.6.1 leads to the following problem scenario. A producer has the latest version of Farm Smart 2000 which contains system data up to and including 1999. He has completed his farming operations for 1999 and is planning ahead for the 2000 growing season by entering crop planning scenarios in Farm Smart 2000. The producer specifies the year as 2000 in tier-3, and as described in Section 6.4.3.6.1, the system data for 1999 are displayed (Its validity range is artificially extended into the future as described in Section 6.4.3.6.1.) The producer enters data for several crops, with their varieties, and specifies some probable herbicide and insecticide applications. The producer, by fine tuning his inputs and running a number of possible

²⁰ The system data will not likely be extended back any further than it currently is as most users will not likely go any further back in entering historical records, as historical data further back will not have any influence on Farm Smart 2000 results for future planning.

scenarios, develops a proposed crop plan for the 2000 growing season for his farming operation. The producer saves several of the scenarios he has developed for later use if desired, and commits the data from his proposed crop plan to the SmartRKS, such that, when he actually executes his plan in the 2000 growing season, he need only supply pertinent dates and any additional information.

Early in year 2000, this producer purchases the new updated version of Farm Smart 2000 which contains updated system data for 2000. In this system data, one of the crop varieties that the producer was planning to seed is now de-registered, as is one of the herbicides he was planning to apply. Consequently, he now has several input data records in the SmartRKS that point at system data that no longer exists²¹. In addition, he has one or more crop planning scenarios that now reference non-existent system data. The next time the producer displays the affected records, the data fields will be blank, according to SmartRKS's operational specifications. If the producer attempts to save this record, an error message will be displayed stating that he must make a selection for that data field. (The affected data fields in these folders must have valid entries, that is, the record cannot be saved if data fields are blank.) Furthermore, the next time the producer attempts to load the affected scenario, he will be informed that some data has been deleted from SmartRKS, and therefore, the affected parts of the scenario will also be deleted. (The data in the scenario file will not actually be deleted until the producer again saves the scenario.) Lastly, the invalid crop variety will now result in an entire column missing from the output selection grid in tier-3.

According to operational specifications, the high end of the system data's validity range has been reset from "forever" to 1999. Hence, where the record was originally valid from the start year to "forever", it is now only valid from the start year to 1999. Therefore if the producer is viewing data from within the scope of year 2000, the record no longer exists.

A possible solution to this problem was to provide clients with timely updates of the system data, but often new data is not available until March or April, which is too late to be of value for planning purposes within the current growing season. Further complicating

²¹ "No longer exists" actually means that for the year and identification associated with the input data fields, system data records can no longer be found.

this matter was the issue related to an associated fee (i.e. price) for system data updates. If there is to be a fee, some users may opt not to purchase the system data updates and thus continue to use old system data, further complicating this problem. If there is no fee associated with the system data updates, it is desirable, from a business perspective, to make the data updates independent of future program improvements. This could limit the ability or flexibility in the type of improvements that could be made, or it may be necessary to supply updated system data in a variety of formats. Thus, it was essential to provide users with the functionality to work ahead, as described in the Section 6.4.3.6.1, but without causing them to lose data.

6.4.3.6.3 Redesign Possibilities

To solve this particular data integration problem, several alternatives and solutions were considered, but the number of feasible solutions were limited. One possible solution was to redesign the operational specifications described in the Section 6.4.3.6.1 such that this problem was no longer an issue. Otherwise, an alternate solution was required which is described below.

Before discussing the solution implemented, an aspect regarding the redesign of the operational specifications will be briefly addressed. One obvious and drastic change to the operational specifications that would have solved this problem would be to not artificially extend year ranges into the future. This would completely solve this data integrity problem, as it would be impossible to work ahead due to there not being any system data whatsoever for the users to work with. However, this was undesirable, as it further increases the pressure to distribute updated system data as soon as possible, which is not completely controllable since the majority of updates, such as variety and herbicides come from elsewhere, increasing the possibility of late updates, resulting in frustrated users, and potential loss of credibility to Farm Smart 2000. Thus, this was not a viable solution.

After a producer gains some experience with SmartRKS, they soon desire “free-form data entry” which enables a user to enter data which is not contained in a system-data pick list, and to use names that may be more meaningful to the user. Although “free-form data entry” would be trivial to implement in SmartRKS, this change would have an enormous impact on the reasoning process in tier-3 in that another higher level of artificial

intelligence would be required in order for tier-3 to be able to reason about the user's ambiguous "free-form data entry" in conjunction with the system data, knowledge-bases, and reasoning processes. "Free-form data entry" is a large undertaking and is referenced in Section 8.7.

6.4.3.6.4 Problem Analysis

A workaround was the only remaining viable solution. Some of the obstacles that had to be considered to implement a workaround included:

- 1) Given the nature of how the system handles obsolete system data as described above, it is impossible for the users to even see what their original data was in order to intelligently select a new option;
- 2) As described previously, there are two main modules in Farm Smart 2000 which were affected by this data integration and work ahead problem, the crop planning module and SmartRKS, and any additional modules that might be added later which could be affected. It is never known which of these modules the user might run first after a system data upgrade. All system- and user-data access is accomplished through the RksAcces DLL, which uses RksUpdate.DLL to perform any necessary data conversion on the user data tables. There has been no requirement for either DLL to have a user-interface, other than a few message boxes in the case of errors;
- 3) The only opportunity to verify a Farm Smart 2000 scenario was when the user accessed it, as it was not practical to search the user's hard drive for all scenario files and attempt to repair them;
- 4) There were a number of opportunities where the SmartRKS input folders could be verified: from the installer, when RksAcces is first initialized (by whatever client application the user executes first), or when the user actually views/uses an affected record, which will depend on the client application being executed and the function being performed (e.g. viewing a record in a folder versus generating a report in SmartRKS);
- 5) It was clear that any problems with Farm Smart 2000 scenarios are unique to Farm Smart 2000 and needed to be solved by that application;

- 6) Any problems with SmartRKS input folders were common to all applications which use SmartRKS. In order to ensure consistency and non-duplication of effort/code, it was desirable to solve these problems in RksAcces, relying on RksUpdate to do some of the work;
- 7) It was possible to find the obsolete system data that is being referred to by searching for the latest year for which a particular system data ID is still valid and using that record. This was possible because when the system data is updated, records are never deleted, but rather the year ranges of existing records are adjusted and new ones are added. One exception to this rule is if two updates of the Prairie Crop Protection Planner data are released in one year and both updates are shipped to customers. Then, it is possible that a product in the first update would be added and then deleted in the second update. In this case, the system data record in SmartRKS would be fully deleted.

6.4.3.6.5 Possible Solutions

There were several possible solutions that were considered for this particular data integration and work ahead problem. If it was a viable alternative to solve the SmartRKS input folder problems in RksAcces and RksUpdate, then an appropriate user interface would need to be designed and developed. But RksAcces and RksUpdate are DLL's that could be used by any number of client programs, and as such, they should not have their own user interfaces. Alternatively, SmartRKS could be used to repair the problems but this would elevate SmartRKS beyond the role of "just another client" and would also make RksAcces dependent on SmartRKS, which results in a design flaw, since SmartRKS already needs to depend on RksAcces. (Dependency cycles such as this are a very poor design idea.) An alternate possible solution was simply to report all data problems found, or interactively walk the user through them, providing the option to make a new choice or delete the affected record.

Another solution considered was to ignore the problem at the RksAcces level and make the client programs responsible for resolving the problems using their own interface. Possibly some infrastructure in RksAcces could help simplify the problem, such as a

function that could look up the original system data that was used when the record was created. But if the first use of an affected record is a report, then the problem data has already been used unnoticed.

6.4.3.6.6 The Solution Implemented

The solution implemented, and described below, for repairing data inconsistencies in SmartRKS, represents the best solution in terms of ensuring data integrity while providing the users as much control as possible over their data. The repair is performed the first time any client program of RksAcces.DLL is executed after the system data upgrade has been performed. No data updating is done during the system data installation. First, in the user's records, the version table keeps track of both the user-data version number, as was the case, and the system data version number, which was added.

When RksAcces initializes, after any client program such as crop planning or SmartRKS is initiated, the user data version identifier is checked against the version identifier it is designed to work with. If the user data version identifier indicates it is older, RksUpdate.DLL is loaded and it converts the user's data to the latest version. When it is done, it updates the version identifier in the user's data.

Similarly, the system data version is checked against the version it is designed to work with. If the system data version is older, a new program/DLL is loaded to check all of the tables that might be affected by the workahead problem. For each record, this program attempts to look up the necessary information in the system data. If unsuccessful, this program invokes a user interactive session to:

- 1) show the farm, years that it was acquired and disposed, and validity range for the record. The system object, that this record used to refer to, is found, and enough information is displayed about it for the user to identify. For example, crop, seed name, and variety data will be shown.
- 2) describe the problem data to the user, for instance:

The variety/herbicide/fungicide/insecticide X is no longer registered for
<year acquired>.

- 3) prompt the user to choose a solution. For example, to repair this inconsistent data, the options are:

- a) Retain this variety by changing the year acquired to <latest year that variety existed>;
- b) Choose another variety/herbicide/fungicide/insecticide: <drop list containing choices>;
- c) Delete the record and all events that use it.

Once the user has made a choice, the program/DLL repairs the current record as requested and then moves on to the next record. The option to change the year acquired could not be presented, as this action would implicitly delete some of the user's data. For example, if the year acquired was 1995 and it needed to be changed to 1998, all events based on the record for 1995, 1996, and 1997 would need to be deleted.

A similar approach was implemented for handling crop planning scenarios, with some minor variations in the interaction to correspond to the respective differences. Tier-3 had to handle situations previously where the user had deleted information from SmartRKS since a scenario was saved. It previously dealt with this situation by simply warning the user and then deleting the affected part of the scenario. This new program was used wherever possible to rectify data inconsistency problems. For instance, if the user had deleted a farm, there was no corrective action that could be invoked; the user is prohibited from loading the scenario. Similarly, if the user had deleted a field, there was little corrective action that could be taken, that is, part of the scenario was deleted. Although an alternative would be to prompt the user to select a different field, a new field would affect the remainder of that part of the scenario, and therefore was best handled by deleting it. The other instances where this aspect arises involves equipment selections, where in these cases the user is allowed to choose a different piece of equipment, thus preserving that part of the scenario.

6.4.3.7 Rule Management Conventions

Most of the rules for Farm Smart 2000 are common to all crops although there are some crop-specific rules. In order to perform the rule-based inference, data is passed to the IRE. For each crop/location combination, the rule base is executed once. The rules are designed to work under the assumption that data from the folders are volunteered to the IRE. Using the data provided, the rule base attempts to produce as many results as

possible. This approach to inference is data-driven or forward-chaining. To ensure that all applicable rules are considered for each crop during an inference process, the first condition of every rule-set in the knowledge-base is a test of the crop name to determine if it is supported.

6.4.3.8 Installer

The installation software (i.e. installer) for Farm Smart 2000 was developed using the *Wise Installation System*[™] and installs the entire Farm Smart 2000 system, including SmartRKS and all the agents. For the Crop Protection Planner and Weed Management Planner, this means duplicating the functionality of their stand-alone installers and managing the consequences of duplicated installations. To avoid undue maintenance problems that this would cause, the following options were considered:

- Provide separate install disks for the agents and require the user to install them either before or after installing Farm Smart 2000. This is excessive work for the user and should not be necessary;
- Provide separate install disks for the agents and have the Farm Smart 2000 installer execute the install files on these disks automatically. The *Wise Installation System*[™] provides nominal support for this, but all installers must have different names (i.e. they could not all be named setup.exe). The agents' installer files could be renamed to make them unique. The individual agents would also need to be uninstalled when Farm Smart 2000 is uninstalled which could be accomplished with the *Wise Installation System*[™]. This approach was not used for the following three reasons:
 - 1) When called as a sub-process of an installation, the *Wise Installation System*[™] does not properly execute multi-disk installers;
 - 2) Users may be confused as there is no progress indication during the installation of agents; and
 - 3) Many files would be installed several times (particularly ODBC).
- Include the agent install disk images in the Farm Smart 2000 installer and place them in a temporary directory, from where they could be executed. This approach would solve the first problem above, but the concerns of progress

indication and duplicate file problems would still exist. The file naming issue could be accomplished more easily with the Farm Smart 2000 installer performing the necessary rename. Additional disk space on the target machine would also be required during the installation, as space would be needed for the temporary disk images:

- Include a "stripped down" version of the agent installers, with the Farm Smart 2000 installation software, which would not include ODBC or common files. This approach could be accomplished by using *Wise Installation System*[™] compiler variables and conditional compilation. This was the best approach and although it meant modifying the agent installation software, these modifications were much easier to maintain than duplicating the entire agent installer scripts in the Farm Smart 2000 installer.

6.4.3.9 Agent Development Guidelines

The Farm Smart 2000 System is an open architecture decision support system that permits new agents to be included with little effort, although agents must follow certain guidelines in order to integrate with Farm Smart 2000 (i.e. tier-3). Agents may execute interactively or non-interactively, or both. Since the tier-3 analysis/evaluation process should not require any user interaction, agents executing in interactive mode should interact with the user only during data entry. In non-interactive mode, the agent is used only for the computations or data it can provide during tier-3's analysis/evaluation phase. In order to execute, agents might require system data, which can be obtained from the SmartRKS database using the RKSAccess DLL.

Farm Smart 2000 communicates with agents by calling functions within a DLL, or via DDE. The developers of the agents can decide which method is appropriate. Details of this communication are outlined in the following sections. Farm Smart 2000 itself (e.g. the Global Control Expert) must be modified in order to support new agents.

6.4.3.9.1 Agent Communication with SmartRKS

If an agent requires system data that is not included in the tier-3 folders, it can access this data directly from SmartRKS using a DLL that implements the RKSAccess API. During tier-3 communication, an agent must always obtain system data from SmartRKS

even if its own internal native data files contain similar information. This ensures that all Farm Smart 2000 analyses are performed using a consistent data set.

If the agent is an executable program, and not a DLL, it should accept the command line switch “/RKS”, that instructs it to access system data from SmartRKS. Optionally, the agent can implement a DDE Execute command that instructs it to use SmartRKS. DLL type agents must support one of these two methods for accessing system data from SmartRKS.

6.4.3.9.2 Agent Communication within Farm Smart 2000

At the basic level of communication, tier-3 uses agents in order to access their local data. This data could be stored within the agent’s internal database(s), or it might need to be calculated by the agent. If calculation is required, the agent must accept a command that instructs it to perform the calculation. If the calculation requires data contained in tier-3 folders, this data can be passed from to the agent either prior to the command or as arguments in the command.

When an agent is processing commands in interactive mode, the user must interact with the agent’s own user interface to complete the command. When processing commands in non-interactive mode, the agent does not need any input from the user. A combination of interactive and non-interactive commands can be processed by an agent that supports multiple commands.

6.4.3.9.2.1 Interactive and Non-Interactive Agents

When an agent executes in non-interactive mode, its main window should be “hidden” from the user. That is, no button should appear in the Windows task bar. However, it is also acceptable to execute minimized. Although non-interactive commands are more often used during the analysis/evaluation phase, they can be used to assist tier-3 during the data-entry stage.

In interactive mode, the user interacts with the agent’s user interface. Any data that is contained in SmartRKS or tier-3 folders must not be user requested again by the agent. However, the agent may display such data in editable form, but it must initially default to the values stored in SmartRKS or tier-3 folders. If the agent requires only a subset of data already stored in SmartRKS or tier-3, it may prompt the user to select this subset. In interactive mode, there must be a method for the users to terminate interaction with the

agent and return to tier-3. Optionally, the agent can have the capability to save or cancel the results of the interaction.

6.4.3.9.2.2 Dynamic Data Exchange

To support agent communication via DDE, the agent must implement a DDE server that supports at least one DDE Execute command (i.e. Quit) which permits tier-3 to stop the agent when communication is complete. The agent must accept a “/DDE” command line switch, and when provided, the agent should start in a hidden or minimized state. If the agent receives a command that requires user interaction, it can then display its user interface. When the agent receives the Quit command, it should stop itself. When the agent receives the Cancel command, it should discontinue the DDE communication, but should not stop itself.

When using DDE communication, data is provided to tier-3 by utilizing the DDE Request or DDE Advise functions. Thus, the agent simply stores the data in a DDE Item, and tier-3 is responsible for accessing this data. If the agent provides native data to tier-3, it can be stored immediately in the appropriate DDE Item. If the agent must first calculate, these calculations should be requested by a DDE Execute command. Any tier-3 folder data required by the calculation can be passed as arguments to the DDE Execute command, or can be stored in DDE Items by tier-3 prior to the DDE Execute request using the DDE Poke function. If more than four or five data items are required, the DDE Poke function should be used.

If a calculation is likely to cause a DDE Execute request to time out (i.e. fail), usually during an interactive command, a different function needs to be implemented. One solution to time out failure is to define a DDE Item named Status and when the agent receives the DDE Execute request, it should set the Status to “Pending” and immediately return from the DDE Execute request. Then once the calculation is complete, the Status Item can be changed to a status such as “OK”, “Cancel”, or “Fail”. With tier-3 monitoring the Status Item via the DDE Advise function, it will obtain the calculation results only after the Status has changed from “Pending”.

Based on these guidelines, a typical DDE communication between tier-3 and an agent might proceed as follows:

- If the agent is not already running, tier-3 will trigger it, passing it the /DDE command-line switch;
- If the agent is not performing calculations, tier-3 will access the required data via the DDE Request function, and send the DDE Execute [Quit] command;
- Otherwise, tier-3 will DDE Poke any data required by the agent;
- Tier-3 sends a DDE Execute command requesting the calculation it wants the agent to perform, passing any necessary arguments to the command;
- The agent completes the calculation and indicates its done (i.e. by returning from the DDE Execute command, or by setting a Status Item as outlined above).
- Tier-3 uses DDE Request to access any results and sends the DDE Execute [Quit] command.

Note that the DDE Execute [Quit] command will only be sent if the agent was triggered by tier-3. If the agent was already executing, it will be left executing. It is the agent's responsibility to restore its previous state after the DDE communication.

6.4.3.9.2.3 Dynamic Link Library

Instead of using DDE, an agent can be implemented as a DLL, providing the necessary functionality to tier-3. If the agent can also execute as a stand-alone application, it should have a separate executable interface program that uses the same DLL.

All communication between tier-3 and a DLL agent utilize function calls. The DLL must provide functions that permit tier-3 to pass the necessary values (similar to DDE Poke above), to initiate calculations (similar to DDE Execute above), and to retrieve results (similar to DDE Request above). A DLL is not a stand-alone program, and thus a DLL agent does not need to be "triggered" by tier-3 nor does it need to support a "Quit" command.

6.4.3.10 Agents and Knowledge-Bases

The following sections describe the agents and knowledge-bases used in tier-3, how the agents are invoked (i.e. DDE type agent or a "triggered" stand-alone agent) and examples of the types of data they provide. The crop yield model (i.e. knowledge-base) is also discussed.

6.4.3.10.1 Crop Variety Select

CVSShell is a Crop Variety shell agent that provides knowledge and information to tier-3 regarding crop varieties. CVSShell is a DDE type agent consisting of an executable program and a database. The /DDE command line argument must be passed to it in order to ensure it runs hidden. The following is a sample of the types of data that CVSShell provides to tier-3.

Value: Days to Maturity

Purpose: Provides the average number of days required for a particular crop variety to mature.

Value: Variety Average Yield

Purpose: Provides the average yield obtained by a particular crop variety.

Value: Variety Type

Purpose: Provides the variety type of a particular canola variety.

6.4.3.10.2 Crop Rotation Planner

RPShell is a shell agent that provides knowledge and information to tier-3 regarding crop rotations. RPShell is a DDE type agent consisting of an executable program and a database. The /DDE command line argument must be passed to it in order to ensure that it runs hidden. The following is a sample of the types of data that RPShell provides to tier-3.

Value: Number of Diseases

Purpose: Provides the number of potential disease problems resulting from a given crop rotation.

Value: Number of Herbicides

Purpose: Provides the number of potential herbicide problems resulting from a given crop rotation.

Value: Maximum Risk

Purpose: Provides the maximum risk of yield loss from any of the potential disease problems. "H", "M", and "L" mean high, moderate, or low respectively. " " means no risk. "F" is equivalent to "H", but also indicates that a field bioassay is required before seeding the given crop.

Value: Name

Purpose: Provides the name of the problem disease or herbicide for which information was requested.

Value: Risk

Purpose: Provides the risk of yield loss resulting from the problem disease or herbicide for which information was requested. See Maximum Risk above for a description of what the values indicate.

Value: Years

Purpose: Provides the number of years for which the requested disease or herbicide could cause problems.

Value: Plus

Purpose: Indicates whether the requested disease or herbicide could cause problems beyond the period indicated by the Years item above. For example, if a disease might cause problems for 3 or more years, Years would be "3" and Plus would be "1". If a herbicide might cause problems for 1 year (and no more), then Years would be "1" and Plus would be "0".

Value: Years Ago

Purpose: Provides the number of years ago the requested potential disease or herbicide problem began.

6.4.3.10.3 Crop Protection Planner

The Prairie Crop Protection Planner is a fully functional agent (i.e. stand-alone expert system) that provides users assistance in planning their crop protection strategies. It consists of modules for herbicide, fungicide, and insecticide applications as well as record keeping facilities. The Prairie Crop Protection Planner is a DDE type agent consisting of an executable program and a database. The /DDE command line argument must be passed to it to ensure that it runs in DDE mode. The following is a sample of the types of data that the Prairie Crop Protection Planner provides to tier-3.

Value: Status

Purpose: Provides the status of the previous operation. The status is "OK" if the operation completed successfully, "Cancel" if the operation was

canceled by the user (during an interactive operation) or by the client application, “Pending” if an interactive operation is still in progress, or “NoSels” if there were no pesticides applicable to the crop and/or pest(s) supplied with the operation.

Value: Pesticide Name

Purpose: Provides the name of the pesticide selected.

Value: Herbicide Group

Purpose: Provides the herbicide group(s) of the selected herbicide.

Value: Actual Rate 1

Purpose: Provides the actual application rate selected by the user. If the pesticide is a tank-mixed herbicide, this is the rate for the first component in the tank mix.

Value: Actual Rate 2

Purpose: Provides the actual application rate selected by the user. If the pesticide is a tank-mixed herbicide, this is the rate for the second component in the tank mix; if the pesticide is not a tank mix or is a fungicide or insecticide, this value is not applicable.

Value: Actual Rate 3

Purpose: Provides the actual application rate selected by the user. If the pesticide is a tank-mixed herbicide, this is the rate for the third component in the tank mix; if the pesticide is not a tank mix, or is a tank mix with only two components, or is a fungicide or insecticide, this value is not applicable.

Value: Recommended Low Rate 1

Purpose: Provides the lowest recommended application rate for the pesticide. If the pesticide is a tank-mixed herbicide, this is the rate for the first component in the tank mix.

Value: Recommended Low Rate 2

Purpose: Provides the lowest recommended application rate for the pesticide. If the pesticide is a tank-mixed herbicide, this is the rate for the second

component in the tank mix; if the pesticide is not a tank mix or is a fungicide or insecticide, this value is not applicable.

Value: Recommended Low Rate 3

Purpose: Provides the lowest recommended application rate for the pesticide. If the pesticide is a tank-mixed herbicide, this is the rate for the third component in the tank mix; if the pesticide is not a tank mix, or is a tank mix with only two components, or is a fungicide or insecticide, this value is not applicable.

Value: Recommended High Rate 1

Purpose: Provides the highest recommended application rate for the pesticide. If the pesticide is a tank-mixed herbicide, this is the rate for the first component in the tank mix.

Value: Recommended High Rate 2

Purpose: Provides the highest recommended application rate for the pesticide. If the pesticide is a tank-mixed herbicide, this is the rate for the second component in the tank mix; if the pesticide is not a tank mix or is a fungicide or insecticide, this value is not applicable.

Value: Recommended High Rate 3

Purpose: Provides the highest recommended application rate for the pesticide. If the pesticide is a tank-mixed herbicide, this is the rate for the third component in the tank mix; if the pesticide is not a tank mix, or is a tank mix with only two components, or is a fungicide or insecticide, this value is not applicable.

Value: Units 1

Purpose: Provides the units for the application rates (actual, low, and high) for the pesticide. If the pesticide is a tank-mixed herbicide, these are the units for the first component in the tank mix.

Value: Units 2

Purpose: Provides the units for the application rates (actual, low, and high) for the pesticide. If the pesticide is a tank-mixed herbicide, these are the

units for the second component in the tank mix; if the pesticide is not a tank mix or is a fungicide or insecticide, this value is not applicable.

Value: Units 3

Purpose: Provides the units for the application rates (actual, low, and high) for the pesticide. If the pesticide is a tank-mixed herbicide, these are the units for the third component in the tank mix; if the pesticide is not a tank mix, or is a tank mix with only two components, or is a fungicide or insecticide, this value is not applicable.

Value: Application Units

Purpose: Provides the units to which the pesticide is applied. For herbicides and insecticides, this will be either acres or hectares. However, for seed and storage treatment fungicides, the application units will vary.

Value: Cost

Purpose: Provides the cost per application unit for the pesticide application.

6.4.3.10.4 Weed Management Planner

The Weed Management Planner is a fully functional agent (i.e. stand-alone expert system) that assists the producer in managing weeds, and includes record keeping support, economic thresholds, problem weed information, long-term weed management, and weed density maps. The Weed Management Planner is a DDE-type agent consisting of an executable program and a database. The /DDE command line argument must be passed to it to ensure that it runs in DDE mode, rather than interactively. The Weed Management Planner, with its six interrelated modules, can provide large amounts of information, too many to give examples of here. However, the Weed Management Planner has an extensive problem weed information file which Farm Smart 2000 tier-3 accesses and uses for various problem-solving activities.

Value: Problem Weed Information File.

Purpose: Provides the name of a problem weed information file.

6.4.3.10.5 Crop Yield Model

Farm Smart 2000 tier-3 integrates a knowledge-base in the form of a yield model to aid in determining possible yield given numerous input factors including available water

and nutrients. The model is integrated in Farm Smart 2000 tier-3 as an executable module. An alternative was to implement the yield model (i.e. knowledge-base) directly in the main knowledge-base (i.e. rule base). However, the original yield model was implemented with the *Stella*® modeling software and uses different units than the rule base. To overcome this incompatibility, it would have been necessary to either convert the entire *Stella*® model into the units used by the knowledge-base, perform unit conversions within the rule base, or pass the inputs in the proper units as expected by the yield model. The latter option would have required two copies of many input values; one in the units expected by the yield model knowledge-base and one in the units expected by the main knowledge-base. All of these approaches introduce potential confusion and subsequent maintenance problems into the knowledge-base.

A major design goal for RKSAcces was that it be language-independent such that any language that can access a Windows DLL should be able to use the RKSAcces API. Hence, for this reason, all public data structures are defined as C structs, which can easily be ported to other languages, and the functions are all global functions (i.e. not class members) declared with C linkage (i.e. using extern "C").

6.4.4 Other Similar Multi-Agent Problem-Solving Systems

The Interrelated Decision Support component of Farm Smart 2000, with its multiple heterogeneous agents integrated into an open architected cooperative problem-solving environment such that crop planning and other related problem solving can be accomplished, is leading edge technology. Although similar research and development has been undertaken in the medical and space technology fields, this technology has not been attempted before in an agricultural application.

Chapter 7: Conclusions

The research resulted in the Farm Smart 2000 agricultural decision support system being designed, developed, and tested. This system combines information, knowledge, and human expertise, through the integration of expert systems, resulting in an aid for the management decision-making process through interaction with the user. It consists of three tiers of progressively more detailed and on-farm specific problem-solving capabilities that provide end-users (e.g. producers) with effective decision support for adopting and maintaining diversified farming operations with a bias for conservation farming systems.

Initial results and responses from SmartRKS beta testers (i.e. producers and agricultural extension specialists) are positive and indicate that Farm Smart 2000 fulfills the apparent need for on-farm decision support. This research concludes that it is possible to provide technology transfer in the complicated subject area of conservation farming using computerized decision support systems. This chapter describes the achievement of the goals and objectives of Farm Smart 2000 as stated in Section 4.3 and Farm Smart 2000's potential for technology transfer and adoption.

7.1 The Achievement of Farm Smart 2000's Goals and Objectives

The first goal was to provide cooperative multi-agent problem solving by integrating the knowledge and expertise of several human agricultural experts and to simulate these experts working cooperatively to solve broad-based agricultural problems. This goal has been achieved by the development of expert systems (i.e. agents) which mimic human agricultural experts. The integration and cooperation of these agents for problem-solving can be demonstrated by watching the *Windows*TM task bar when Farm Smart 2000 is processing an evaluation. During this evaluation process, every time an agent executes and contributes a partial solution to the blackboard, the agent's name will appear in the *Windows*TM task bar. Since the *Windows*TM task bar shows what task is executing at any one time, during the evaluation process, the agents' names (i.e. task name) will be displayed many times as each agent takes its turn in providing partial solutions to

the problem. Another indicator that there are multiple agents cooperating to solve problems is that the result folders contain results from different specialized areas (i.e. provided by more than one agent) (See Figure 5.2).

The second goal was to develop an open system architecture allowing for new agents to be integrated without re-engineering the system each time. As described in Section 1.3, this was a goal from the onset of the project and has been continually met with the integration of each new version of the Crop Protection Planner. The Crop Protection Planner expert system/agent has evolved through two major different versions during the development of Farm Smart 2000, and each time this new expert system has been successfully integrated into the open system architecture. One of the new versions of the Crop Protection Planner was the result of incorporating the Herbicide Planner code into the Crop Protection Planner to include the capabilities of the Herbicide Planner.

The third goal was to provide decision support that is specific to individual farming systems (i.e. farm-specific decision support). The achievement of this goal is evident from the detailed farm-specific results obtained in the result folders. However, this is accomplished only when the user provides farm-specific data to Farm Smart 2000 via the SmartRKS record keeping system and the tier-3 input folders. It has been clearly described throughout the dissertation that the SmartRKS record keeping system is the major contributor to accomplishing this goal.

7.2 Technology Transfer and Adoption

The requirement to substitute knowledge for labour (e.g. through technology transfer) continues to intensify and decision support systems will play a major part in fulfilling this requirement. Economics has, and will continue to play a major role in matching the correct cropping systems to the available land base. Decision support systems to support these complex diversified cropping systems cannot be developed in isolation, but rather must have the collaborative effort of many so that knowledge can be integrated. Farm Smart 2000, with its open architecture design, was developed to accomplish this need by integrating individual heterogeneous expert systems in a cooperative problem-solving system. It is "open" to other issues by adding, deleting and/or updating knowledge sources and agents without re-engineering the entire system, thereby increasing its

technology transfer abilities, decision-support capabilities, and its credibility, beyond what a single human agricultural extension specialist can offer.

Farm Smart 2000 is a vehicle or mechanism for ongoing transfer of technology and knowledge from the research laboratories and field plots to the end-user. In so doing, Farm Smart 2000 provides "single-window" access to information and knowledge at any time. Farm Smart 2000 is accurate because it is farm specific, utilizing the producer's own data when analyzing solutions and making recommendations. The strength of Farm Smart 2000 is diversity, with its many various knowledge sources, not just one, resulting in a holistic system. Farm Smart 2000 avails of credible knowledge and information from various sources including human experts, published material, research data and results, and then transforms it into "decision support", delivering it to the end-user. By adding value to the producer's data, Farm Smart 2000 helps the producer make decisions or provides choices that are specific to the producer's own farming operation.

Public extension and community colleges have a role in teaching producers how to benefit from the use of electronic information technology. Extension has always been involved in the adoption of new production technology. However, it is a relatively new concept to transfer computer technology to producers. It is believed that with the completion of Farm Smart 2000, public and private extension will play an important role in the continuance of field testing the system, at which time the importance of decision support systems in technology transfer and extension programming will be fully understood.

7.3 Contributions of Research

The research, design, and development of Farm Smart 2000 has furthered the science and technology of decision support systems. Lander (1994) states that an agent can be either an existing software (e.g. "off-the-shelf"), modified to work within an agent set, or it can be software specifically implemented as a reusable agent to work within an agent set. Redesigning agents to work within an agent set is more problematic. Neches et al. (1991) indicate that sharing and reusing knowledge sources in any form is difficult, but feasible, if knowledge sharing technology and infrastructure tools can be created with which to facilitate knowledge-based system development and operation. This research was

challenged with the integration and utilization of heterogeneous reusable intelligent expert systems to form a cooperative problem-solving system.

Furthermore, Nii (1986) specifically points out that the blackboard model does not specify how a blackboard is to be realized as a computational system. That is, Nii (1986) states the blackboard model is a conceptual entity, not a computational specification. In addition, Nii (1986) informs us that given a problem to be solved, the blackboard model provides sufficient guidelines for sketching a solution, but a sketch is far from a working system. To design and develop a system, a detailed model is required.

In this research of cooperative problem solving, a top-level design for Farm Smart 2000 was developed and implemented based on the integration of heterogeneous reusable agents through the use of conflict management techniques (i.e. PARI DSS '95 prototype). This prototype was limited in that it could not adequately represent the complex interrelations of conservation farming systems, resulting in only general decision support. The development of Farm Smart 2000 ascertains that a bottom-up design utilizing the blackboard model and rule management techniques in conjunction with active object-oriented database technology can accomplish the collaboration and coordination required for the integration of heterogeneous reusable agents in a cooperative problem-solving environment.

The program coding of expert systems, for integration in Farm Smart 2000, began in 1993, with the beta version of Farm Smart 2000 being completed in 1999. To maintain schedules for completion and release of expert systems to producers, the methods and technology pertaining to decision support systems that were available in the late 1980s and early 1990s were considered and exploited in the software development of Farm Smart 2000. Although additional and different methods and techniques (not necessarily better) have been researched more recently, they have not been considered for use in Farm Smart 2000 because of the changes to the overall architecture and framework that would be necessary, resulting in slowing down development.

Sycara (1998) reports that agent-based systems technology continues to be popular as an approach for conceptualizing, designing, and implementing software systems. Agent-based systems remain particularly attractive for developing software that operate in open

environments. Sycara (1998, p.79) states that "Currently, the great majority of agent-based systems consist of a single agent.", unlike Farm Smart 2000 which is a multi-agent system. Sycara (1998, p.79) also states "However, as the technology matures and addresses increasingly complex applications, the need for systems that consist of multiple agents that communicate in a peer-to-peer fashion is becoming apparent.". The characteristics of the multi-agent systems that Sycara (1998) refers to include:

- 1) each agent has insufficient knowledge for solving the problem;
- 2) there is no global system control;
- 3) data are decentralized;
- 4) computation is asynchronous.

The Farm Smart 2000 multi-agent system has the same characteristics as stated in 1) and 4). Farm Smart 2000's characteristics differ with 2) above in that it has a Global Control Expert to provide global system control. Although Farm Smart 2000's data are also decentralized to a degree, as indicated in 3) above, it does have a global data storage area (i.e. blackboard database). It can be argued whether global system control (i.e. Farm Smart 2000's Global Control Expert) or peer-to-peer communication is superior. However, one important aspect to consider is the degree of "openness" of the agents' environment. If for instance, the open environment is the Internet, where agents dynamically appear and disappear, then there is no opportunity for global system control.

Furthermore, Sycara (1998) indicates that the issues and challenges for the development of multi-agent systems include:

- 1) the formulation, description, and allocation of problems, and the synthesizing of results among a group of intelligent agents;
- 2) enabling agents to communicate and interact, and heterogeneous agents to interoperate;
- 3) ensuring that agents act coherently in their decision making and actions;
- 4) enabling individual agents to reason about the actions, plans, and knowledge of other agents to establish coordination among them;
- 5) resolving different viewpoints and conflicting intentions.

Although Farm Smart 2000's open environment is less complex, where agents do not dynamically enter and exit, Farm Smart 2000 solves all of these challenges nonetheless.

Sycara (1998) indicates that in these complex open environments, researchers are investigating "middle agents" that maintain knowledge of other agents, similar to meta-knowledge that is utilized in Farm Smart 2000. When an agent is triggered, it advertises its capabilities to a middle agent that helps locate another agent with a particular desired capability.

A research area that might benefit Farm Smart 2000 is conversational agents, which are agents that converse with humans while solving problems [Allen, 1998]. These forms of interaction could support diagnosis, design, planning, scheduling, information retrieval, command and control, and other task-directed activities [Jones et al., 1999].

As discussed in this dissertation, much research has been devoted to developing practical techniques for achieving agent coordination. The author developed a Global Control Expert to accomplish the required coordination within Farm Smart 2000's open architecture. In agent-to-agent communication environments, another research area of interest is the notion that the amount of agent meta-knowledge required for coordinating agent interaction, outstrips the agent's limited reasoning capacity (e.g. available time, memory) [Durfee, 1999]. Some of the meta-knowledge required in a peer-to-peer relationship includes agents knowing about themselves, about other agents, and about how other agents view themselves and others, which can amount to excessive meta-knowledge.

Other references and research that could be considered beneficial to Farm Smart 2000 include:

- 1) A book titled "Multiagent Systems, A Modern Approach to Distributed Artificial Intelligence" edited by G. Weiss containing chapters by familiar authors (i.e. to this research). Some chapters of interest include: "Multiagent Systems and Societies of Agents" authored by M.N. Huhns and L.M. Stephens; "Distributed Problem Solving and Planning" authored by E.H. Durfee; and "Computational Organization Theory" authored by K.M. Carley and L. Gasser [Weiss, 1999].

- 2) A book titled “Object Database Development, Concepts and Principles” by D.W. Embley which discusses the principles and concepts required for developing advanced database applications and how to apply them successfully. Although Embley’s (1998) development approach is object oriented, it allows for a broad range of target database systems including standard relational database systems, object-relational database systems, object-oriented database systems, and active database systems [Embley, 1998].
- 3) A book titled “The Object Data Standard: ODMG 3.0” edited by R.G.G. Cattell and Douglas K. Barry. This book provides all the details comprising the Object Data Management Group (ODMG) 3.0 and is the latest version of specifications. It discusses the newest methods for storing objects in databases for the development of object database products [Cattell and Barry, 2000].

Chapter 8: Further Opportunities for Research and Development

Although Farm Smart 2000 is presently a working decision support system, there is room for enhancements and improvements which are discussed in this Chapter.

8.1 Summarize Results

Result summaries would be an asset to improve the clarity and understandability of the information that Farm Smart 2000 provides. This would probably require the development of an additional knowledge-base with rules to analyze the results for each crop/location combination and then combine and summarize them into a meaningful format.

8.2 Support for Alternate and Specialty Crops

Farm Smart 2000 presently provides support for farm management aspects such as variety selection, planting, crop rotations, fertility, weeds, disease, residue management, harvesting, soil conservation and economics, for crops wheat, canola, barley, oats, and flax. But, Farm Smart 2000 has a limited knowledge-base for alternative or specialty crops. The number of crops supported needs to be increased in order to maximize decision support for diversified crop production, including specialty crops, forage crops, and management of rangeland. Relevant information and knowledge available among individual experts must be sought out, analyzed for its usefulness, organized into appropriate knowledge structures to maintain consistency, and developed into applicable decision processes to fill these gaps in decision support. Again, because of the open system architecture on which Farm Smart 2000 is developed, enhancements like these can be readily implemented.

8.3 Support for Precision and/or Landscape Farming

To enhance Farm Smart 2000 to include decision support for precision and/or landscape farming several enhancements are required which are outlined below:

- 1) The incorporation of spatially related data into Farm Smart 2000 for greater flexibility in representing data and information internally, and to the user, by

modifying and/or expanding the current data structures and knowledge-bases to support the addition of spatially related data;

- 2) Collecting the required agronomic expert knowledge and information and then developing the precision farming rule-sets, knowledge-bases, algorithms, and internal data and knowledge relationships to permit internal processing of decision support results;
- 3) Providing intuitive data entry and presenting results through the integration of LANDBASE²² and digitized maps;
- 4) Incorporating the integration and interaction of Global Positioning System (GPS) and Geographic Information System (GIS) technologies to allow the quantification and mapping of spatial variation. GIS systems are very complex and it would not be required to incorporate full GIS capabilities in Farm Smart 2000, but some of the less complex viewing and map manipulative capabilities are reasonable additions. The ability to import GIS data, store it, and use it in analyses, as well as exporting data to GIS systems, are also reasonable additions. Incorporating the integration and interaction with GPS and GIS technologies in Farm Smart 2000 allows the quantification and mapping of spatial variation. GPS and GIS provide the capability to define and manage much smaller units than the current field size. Variable rate application technology provides the capability to automatically adjust nutrient and pesticide inputs to match the requirements of these smaller management units. The incorporation of spatially-related data would provide greater flexibility in representing data by allowing data input and output to be graphically implemented.

²² LANDBASE is a soil information system which supplies soil resource, land use, and productivity data for the agricultural areas in Saskatchewan. LANDBASE was developed collaboratively by the Department of Soil Science, University of Saskatchewan, and AAFC. LANDBASE is keyed on legal location, and hence, using the producer's legal location from their SmartRKS Record Keeping System, which would be graphically linked to a map on the screen, user input could be lessened, with data being retrieved from internal databases with just a click on the displayed map. The resultant soil information would enable Farm Smart 2000 to enhance its references to yields and more.

8.4 Agent Framework

To make it easier to support new agents in Farm Smart 2000, a more flexible agent interaction framework could be introduced. Using such a framework, each agent's meta-knowledge files would be expanded to contain additional information about the agent. For example, how to start it, the operations it makes available, the inputs it expects (in terms of objects in the rule base), and the results it can produce (again, in terms of objects in the rule base). The Global Control Expert would access the agent's meta-knowledge files and use them to add buttons to various input folders to invoke interactive commands when the agents are executing. This would assist the user with data entry and could automatically infer results from inputs by calling the applicable functionality in the agent.

Although this enhancement would be a large undertaking, there are obvious benefits to this approach, in that it would theoretically be possible to add a new agent without modifying Farm Smart 2000. However, in practice, most new agents would require data that doesn't exist in Farm Smart 2000's databases, or would produce results that Farm Smart 2000 cannot currently use in its analysis and calculations. Thus, changes to Farm Smart 2000 in order to support new agents is inevitable.

8.5 Support for Object Linking and Embedding Automation

It would be desirable to permit agents to be invoked and/or controlled via Object Linking and Embedding (OLE) automation. This mechanism would provide all of the functionality that currently exists in Farm Smart 2000 via DDE and some additional capabilities. Furthermore, agents developed with modern methods are likely to support OLE automation "out of the box" and therefore would not have to be customized to work with Farm Smart 2000.

With the use of OLE automation, it should be possible to use the agent's type library to generate or develop code in Farm Smart 2000 to support communications with the agent. The agent would not need to be modified nor conform to Farm Smart 2000's Agent Development Guidelines.

8.6 Support for Graphical Icon User Interface

A graphical icon user interface would greatly simplify data input, making it very intuitive, where farmers would select, for example, icons that look like farm equipment (e.g. tractors and seeders) and farm inputs (e.g. seed, fertilizer, and herbicides), and move them via a mouse to the graphically represented fields to denote the seeding operation as it pertains to reality. When such an action is performed, data would be analyzed “behind the scenes”, such as fuel consumption, seed used, seeding depth, fertilizer used, and herbicides used etc. and then would be stored appropriately with adjusted inventories.

8.7 Free-Form Data Entry

As discussed in Section 6.4.3.6.3, free-form data entry would be viewed favorably by Farm Smart 2000 users enabling them to use names and data that are more meaningful (e.g. crops, varieties, pests, and pesticides) without being constrained to entering data contained in system-data pick lists.

One alternative would be to continue to provide system data, which is user selectable, but also permit the user to enter free-form data. The record, containing the free-form data entry, and any records that are dependent on it, would be flagged as being unusable for the purpose of analyzing decision support (i.e. reasoning) in tier-3. Hence, these records and any records that are invalidated by the updated system data will essentially have their status changed from “usable by tier-3” to being “unusable by tier-3”. The user’s “free-form data entries” will be preserved for the advantages associated with record keeping, but tier-3 will not have the ability to reason about the “free-form data”.

8.8 SmartRKS Compatibility and Import/Export Capabilities

SmartRKS is naturally compatible with Farm Smart 2000 tier-3. It would be desirable to provide agents in tier-2 (i.e. Advanced Decision Support), the same capability to store and retrieve data from SmartRKS (see Figure 5.3), enhancing the capability of “one-time data entry” to essentially all of Farm Smart 2000, rather than just tier-3. It would also be desirable to provide SmartRKS with import/export capabilities to exchange data with 3rd party farm accounting and financial applications which would extend the capabilities of SmartRKS and ultimately Farm Smart 2000’s effectiveness.

References

- Abd-Ella, M.M., E.O. Hoiberg and R. Warren. 1981. Adoption Behaviour in Family Farm Systems: An Iowa Study. *Rural Sociology*. 46:42-61.
- Acton, D.F. and L.J. Gregorich. (Eds.) 1995. *The Health of Our Soils - Toward Sustainable Agriculture in Canada*. Centre for Land and Biological Resource Research. Agriculture and Agri-Food Canada, Ottawa, Ont. XIV, 158 pages.
- Agriculture Museum. 1996. Farming in Canada. [On-line] Available: www.agriculture.nmstc.ca/tour/efarm.htm. Agriculture and Agri-Food Canada.
- Alberta Agriculture. 1983. Information needs of Alberta Farmers and Farm Families. Planning Secretariat, Edmonton, Alberta.
- Alexiev, V., K. Duhaime and P. Dzikowski. 1995. STARRT - Stepwise Technology Adoption Risk Reduction Tool. (v1.3) [Computer software]. Alberta Agriculture, Edmonton, Alberta.
- Allen, J.F. 1998. AI Growing Up. *AI Magazine*. 19(4):13-23.
- Anderson, D.W. 1995. Sustainability of the Semiarid Prairie Ecosystem. [On-line] Available: www.iisd1.iisd.ca/agri/nebraska/show.htm.
- Bailey, K.L., B.D. Gossen and L.J. Duczek. 1995. Impact of Tillage, Rotation, and Location on Control of Plant Diseases. Agriculture and Agri-Food Canada, Saskatoon Research Centre, Saskatoon, Saskatchewan.
- Baker, H.R. 1984. A Proposal for a Canadian Study of Agricultural Technology Transfer/Extension. Unpublished report to the Canadian Agricultural Research Council, Ottawa.
- Barao, S.M. 1992. Behavioral aspects of technology adoption. *Journal of Extension*. Madison, Wisconsin, 30:11-13.
- Barrett, J.R. and D.D. Jones. (Eds.) 1989. *Knowledge Engineering in Agriculture*. The American Society of Agricultural Engineers, St. Joseph, Michigan.
- Barrett, J.R. 1992. Information Requirements and Critical Success Factors for Corn/Soybean Decision Support Systems. pp. 147-152. *In: Proceedings of the 4th International Conference on Computers in Agricultural Extension Programs*. Orlando, Florida.
- Beal, G. and J.M. Bohlen. 1962. How farm people accept new ideas. Special Report No. 15. Cooperative Extension Services, Iowa State University, Ames, Iowa.
- Bentham, M.J., R.A. Coulman and H.C. de Gooijer. in press. PARI DSS Farm Smart 2000 IRDS Road Map. Tech. Report. Agriculture and Agri-Food Canada. Semiarid Prairie Agricultural Research Centre, Swift Current, Saskatchewan.
- Bentham, M.J., R. Coulman and H.C. de Gooijer. in press. PARI DSS Farm Smart 2000 SmartRKS User's Guide. Tech. Report. Agriculture and Agri-Food Canada. Semiarid Prairie Agricultural Research Centre, Swift Current, Saskatchewan.

- Bentham, M.J., R. Coulman and H.C. de Gooijer. in press. PARI DSS Farm Smart 2000 User's Guide. Tech. Report. Agriculture and Agri-Food Canada. Semiarid Prairie Agricultural Research Centre, Swift Current, Saskatchewan.
- Bentham, M.J., C. McInnes, R.A. Coulman and H.C. de Gooijer. in press. PARI DSS Farm Smart 2000 Developer's Guide. Tech. Report. Agriculture and Agri-Food Canada. Semiarid Prairie Agricultural Research Centre, Swift Current, Saskatchewan.
- Bentham, M.J., R.A. Coulman, R. Coulman and H.C. de Gooijer. in press. PARI DSS Farm Smart 2000 Agent Guidelines. Tech. Report. Agriculture and Agri-Food Canada. Semiarid Prairie Agricultural Research Centre, Swift Current, Saskatchewan.
- Berndtsson M. and J. Hansson. (Eds.) 1996. Active and Real-Time Database Systems (ARTDB-95). Springer-Verlag, Berlin, Germany.
- Beynon-Davies, P. 1991. Expert Database Systems: A Gentle Introduction. McGraw-Hill Book Company.
- Blackboard Technology Group, Inc. The Blackboard Problem-Solving Approach. 401 Main Street, Amherst, Massachusetts, 01002. 7 pages.
- Blackburn, D.J. 1986. Technology Transfer. pp. 107-123. *In*: D.W. Anderson (Ed.) In Search of Soil Conservation Strategies in Canada.
- Blackshaw, R.E., F.J. Larney, C.W. Lindwall and G.C. Kozub. 1994. Crop Rotation and Tillage Effects on Weed Populations on the Semi-Arid Canadian Prairies. Weed Technology. 8:231-237.
- Boehm, M.M. 1995. The Long-term Effects of Farming Practices on Soil Quality, as Influenced by Farmer Attitude and Farm Characteristics. doctoral dissertation, University of Saskatchewan, Saskatoon, Saskatchewan.
- Bond, A.H. and L. Gasser. (Eds.) 1988. An Analysis of Problems and Research in DAI. *In*: Readings in Distributed Artificial Intelligence. Morgan Kaufmann Publishers, San Mateo, California.
- Bose, S.P. 1962. Peasant values and innovations in India. American Journal of Sociology. 67:552-556.
- Brown, L.A., E.T. Malechki and A. N. Spector. 1976. Adopter categories in a spatial context: alternative explanations for an empirical regularity. Rural Sociology. 41:99-118.
- Brown, W.J., R.S. Gray and J.S. Taylor. 1994. Economic Factors Contributing to the Adoption of Reduced Tillage/Direct Seeding Technologies in Central Saskatchewan. pp. 13-14. *In*: Proceedings of the Implications of Crop Residue Management and Conservation Tillage Conference. Canadian Society of Soil Science, Regina, Saskatchewan. July.
- Bultena, G.L. and E.O. Hoiberg. 1983. Factors Affecting Farmers' Adoption of Conservation Tillage. Journal of Soil and Water Conservation. 38(3): 281-284.

- Byerlee, D. and E.H. de Polanco. 1986. Farmers Stepwise Adoption of Technological Packages: Evidence from the Mexican Altiplano. *American Journal of Agricultural Economics*. pp. 519-527.
- Campbell, C.A., B.G. McConkey, R.P. Zentner, F. Selles and F.B. Dyck. 1992. Benefits of Wheat Stubble Strips for Conserving Snow Precipitation in Southwestern Saskatchewan. *Journal of Soil and Water Conservation*. 47:112-115.
- Cancian, F. 1967. Stratification and risk-taking: A theory tested on agricultural innovations. *American Sociological Review*. 32:912-927.
- Caprio, J.M. 1986. Potentials for Harvesting Water from Snow. *In: Proceedings of the 1986 Snow Management for Agriculture Great Plains Agricultural Publication No. 120*. Swift Current, Saskatchewan.
- Carlson, J.E. and D.A. Dillman. 1983. Influence of kinship arrangements on farmer innovativeness. *Rural Sociology*. 48:183-200.
- Cattell, R.G.G. and Douglas K. Barry. (Eds.) 2000. *The Object Data Standard: ODMG 3.0*. Morgan Kaufman Publishers, San Francisco, California.
- Chakravarthy, S. 1989. Rule Management and Evaluation: an Active DBMS Perspective. *SIGMOD RECORD*. 16(3): 20-28.
- Chakravarthy, S., K. Karlapalem, S.B. Navathe and A. Tanaka. 1992. Database Supported Cooperative Problem Solving. Tech. Report UF-CIS-TR-92-046. Department of Computer and Information Science, University of Florida, Gainesville, Florida, December.
- Chakravarthy, S., E. Anwar and L. Maugis. 1993. Design and Implementation of Active Capability for an Object-Oriented Database. Tech. Report UF-CIS-TR-93-001. Computer and Information Sciences. University of Florida, Gainesville, Florida, January.
- CIMMYT, 1988. *From Agronomic Data to Farmer Recommendations: An Economics Training Manual*. Mexico, D.F.
- Corkill, D.D., K.Q. Gallagher and P.M. Johnson. 1988. Achieving Flexibility, Efficiency, and Generality in Blackboard Architectures. *In: A.H. Bond and L. Gasser (Eds.) Readings in Distributed Artificial Intelligence*. Morgan Kaufmann Publishers. San Mateo, California.
- Corkill, D. 1991. Blackboard Systems. *AI Expert*. 6(9):41-47.
- Culver, D. and R. Seecharan. 1986. Factors that influence the adoption of soil conservation technologies. *Canadian Farm Economics*. 20: 9-13.
- Cutworth, H.W. and B.G. McConkey. 1997. Stubble Height Effects on Microclimate, Yield and Water Use Efficiency of Spring Wheat Grown in a Semiarid Climate on the Canadian Prairies. *Canadian Journal of Plant Science*. 77:359-366.
- Date, C.J. 1995. *An Introduction to Database Systems*. Sixth Edition. Addison Wesley Publishing Company, Inc.

- Davis, A.M. 1993. Software Requirements. PTR Prentice Hall Publishing, Englewood Cliffs, New Jersey.
- Dayal, U., A.P. Buchmann and D.R. McCarthy. 1986. Rules are objects Too: A Knowledge Model for An Active, Object-Oriented Database System. pp. 129-143. *In*: K.R. Dittrich (Ed.) Proceedings of the International Workshop on OODBs. Springer-Verlag.
- Dayal, U. 1989. Active Data Base Management Systems. SIGMOD Record. 18(3):150-169.
- de Gooijer, H.C. in press. CROPMAN Users Manual. 20-20 Agricultural Services, Kelliher, Saskatchewan.
- de Jong, E., R.G. Kachanoski and B.A. Rapp. 1986. Possibilities for Snow Management in Saskatchewan. *In*: Proceedings of the 1986 Snow Management for Agriculture. Great Plains Agricultural Publication No. 120. Swift Current, Saskatchewan.
- Diaz, O., N. Paton and P. Gray. 1991. Rule Management in Object Oriented Databases: A Uniform Approach. pp. 317-326. *In*: Proceedings of the 1991 International Conference on Very Large Databases. Barcelona, Spain.
- Dickey, E.C., J.C. Siemens, P.J. Jasa, V.L. Hofman and D.P. Shelton. 1992. Tillage System Definitions. *In*: Conservation Tillage Systems and Management - Crop Residue Management with No-till, Ridge-till, Mulch-till. Document MWPS-45. Agricultural Engineering, North Dakota State University, North Dakota.
- Domitruk, D., B. Crabtree, G. R. Coutts and R.K. Smith. (Eds.) 1997. Zero Tillage - Advancing the Art. Manitoba - North Dakota Zero Tillage Farmers Association.
- Duhaime, K., V. Alexiev and P. Dzikowski. in press. STARRT Users Manual, Alberta Agriculture Press, Edmonton, Alberta.
- Durfee, E.H. 1987. A Unified Approach to Dynamic Coordination: Planning Actions and Interactions in a Distributed Problem Solving Network. doctoral dissertation. University of Massachusetts, Amherst, Massachusetts.
- Durfee, E.H., V.R. Lesser and D.D. Corkill. 1987a. Coherent Cooperation Among Communicating Problem Solvers. IEEE Transactions on Computers. 36(11):1275-1291.
- Durfee, E.H., V.R. Lesser and D.D. Corkill. 1987b. Cooperation Through Communication in a Distributed Problem Solving Network. *In*: M.N. Huhns (Ed.) Distributed Artificial Intelligence. Pitman Publishing/Morgan Kaufman Publishers, San Mateo, California.
- Durfee, E.H. 1999. Practically Coordinating. AI Magazine. 20(1):99-116.
- Embley, D.W. 1998. Object Database Development, Concepts and Principles. Addison-Wesley Longman, Inc., Reading, Massachusetts.
- Engelmore, R. and T. Morgan. (Eds.) 1988. Blackboard Systems. Addison-Wesley Publishing Company, New York.

- Ensor, J.R. and J.D. Gabbe. 1988. Transactional Blackboards. *In: A.H. Bond and L. Gasser (Eds.) Readings in Distributed Artificial Intelligence*. Morgan Kaufmann Publishers, San Mateo, California.
- Ervin, C.A. and D.E. Ervin. 1982. Factors Affecting the Use of Soil Conservation Practices: Hypothesis, Evidence, and Policy Implications. *Land Economics*. 58(3):277-292.
- Eswaran, K.P. and D.D. Chamberlin. 1975. Functional Specifications of a Subsystem for Database Integrity. pp. 48-68. *In: Proceedings of the VLDB International Conference*.
- Evans R. and D. Fleury. 1993. Conservation Farming Guide. Alberta Conservation Tillage Society.
- Fayyad, U.M., G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy. (Eds.) 1996. *Advances in Knowledge Discovery and Data Mining*. American Association for Artificial Intelligence, Menlo Park, California and The MIT Press, Cambridge, Massachusetts.
- Feder, G. 1980. Farm Size, Risk Aversion and the Adoption of New Technology under Uncertainty. *Oxford Economic Papers*. 32: 263-284.
- Feder G., and G. O'Mara, 1981. Farm Size and the Adoption of Green Revolution Technology. *Economic Development and Cultural Change*. 30:59-76.
- Fishbein, M. and I. Ajzen. 1975. *Belief, Attitude, and Behaviour*. Addison-Wesley Publishing Co., Reading, Massachusetts.
- Flinn, W.L. and D.E. Johnson. 1974. Agrarianism among Wisconsin farmers. *Rural Sociology*. 39(99): 187-204.
- Fox, G.C., A. Weersink, G. Sarwar, S. Duff, B. Deen. 1991. Comparative Economics of Alternative Agricultural Production Systems: A Review. *Northeastern Journal of Agricultural Economics*. 20:124-142.
- Furtan, H. 1981. Saskatchewan Farm and Household Survey. unpublished data, Department of Agricultural Economics, University of Saskatchewan, Saskatoon, Saskatchewan.
- Gagnon, Y.C. and J.M. Toulouse. 1996. The Behavior of Business Managers When Adopting New Technologies. pp. 59-74. *In: Technological Forecasting and Social Change*. 52(1).
- Garven & Associates 1998. 1998 Western Canadian Benchmark Survey. [On-line] Available: www.agric.gov.ab.ca/economic/mgmt/benchmark_survey/index.html.
- Gill, T.G. 1996. Expert Systems Usage: Task Change and Intrinsic Motivation. *MIS Quarterly*. 20(3):301-329.
- Goldstein, I. 1985. Bargaining Between Goals. pp. 175-180. *In: Proceedings of the 1985 International Joint Conference on Artificial Intelligence*. Tbilisi, USSR.
- Goldstein, B. and R.L. Eichorn. 1961. The Changing Protestant ethic: Rural patterns in health, work, and leisure. *American Sociology Review*. 26:557-565.

- Goss, Gilroy & Assoc. 1985. A Study of Approaches to Improving the Effectiveness of Technology Transfer in Primary Agricultural Production with Specific Emphasis on the Research-Extension Linkage. Report to Agriculture Canada, Tech. Proposal DSS# 54SZ.01B28-5-0184, Ottawa, Ontario.
- Haddadi, A. 1995. Communication and Cooperation in Agent Systems, A Pragmatic Theory. Springer-Verlag, Berlin and Heidelberg, Germany.
- Han, J., Y. Cai and N. Cercone. 1992. Knowledge Discovery in Databases: An Attribute-Oriented Approach. pp. 547-559. *In: Proceedings of the 18th Very Large Database Conference.*
- Heimlich, R.E. 1985. Land ownership and the Adoption of Minimum Tillage: Comment. *American Journal of Agricultural Economics.* 67(3): 679-681.
- Herrmann, V. and P. Uttitz. 1990. If Only I Didn't Enjoy Being a Farmer! *Sociology Rurals.* pp. 62-75.
- Hewitt, C. 1977. Viewing Control Structures as Patterns of Passing Messages. *Artificial Intelligence.* 8:323-364.
- Hewitt, C. 1985. The Challenge of Open Systems. *Byte.* 10(4):223-242.
- Hewitt, C. 1988. Offices are Open Systems. *In: A.H. Bond and L. Gasser (Eds.) Readings in Distributed Artificial Intelligence.* Morgan Kaufmann Publishers, San Mateo, California.
- Hewitt, C. 1991. Open Information Systems Semantics for Distributed Artificial Intelligence. *Artificial Intelligence.* 47(1):79-106.
- Hiebert, D. 1974. Risk Learning, and the Adoption of Fertilizer Responsive Seed Varieties. *American Journal Agricultural Economics.* 56:764-768.
- Hill, W.L. 1964. The Need for Fertilizers. *In: A. Stefferud (Ed.) Farmer's World, The Yearbook of Agriculture.* United States Department of Agriculture, The United States Printing Office, Washington, D.C.
- Holt, D.A. 1989. The Growing Potential of Expert Systems in Agriculture. *In: J.R. Barrett and D.D. Jones (Eds.) Knowledge Engineering in Agriculture.* American Society of Agricultural Engineers, St. Joseph, Michigan.
- Hooks, G.M., T.L. Napier and M.V. Carter. 1983. Correlates of Adoption Behaviours: The Case of Farm Technologies. *Rural Sociology.* 48:308-323.
- Houtsma, J. 1997. Owner's Report on GPS Precision Farming Equipment. *In: Farm Show.* 21(1).
- Hudson, S. and R. King. 1990. Cactis: A self-adaptive, concurrent implementation of an object-oriented database management system. pp. 237-246. *In: Proceedings of the 1990 ACM SIGMOD.*

Hussey, G.A. 1992. Information Technology Adoption by Pennsylvania Cooperative Extension Service: Strategies That Worked. pp. 628-633. *In: Proceedings of the 4th International Conference of Computers in Agricultural Extension Programs*, University of Florida, Orlando, Florida. ASAE publication, St. Joseph, Michigan.

Huth, H. 1957. *Nature and America: Three Centuries of Changing Attitudes*. University of California Press, Berkeley, California.

Ikerd, J.E. 1991. A Decision Support System for Sustainable Farming. *Northeast Journal of Agricultural and Resource Economics*. April. 20(1):109-113.

Jagannathan, V., R. Dodhiawala and L.S. Baum. (Eds.) 1989. *Blackboard Architectures and Applications*. Academic Press, San Diego, California.

Jarvis, A.M., M.E. Rister, W.R. Grant and J.W. Mjelde. 1992. Texas rice producers' technology adoption levels--computers, management, and production practices. Texas Agricultural Experiment Station. (1733).

Jefferson. Jefferson's Moldboard. [On-line] Available: www.monticello.org/matters/interests/moldboard.html. Accessed: February 17, 1997.

Jones, R.M., J.E. Laird, P.E. Nielsen, K.J. Coulter, P. Kenny and F.V. Koss. 1999. Automated Intelligent Pilots for Combat Flight Simulation. *AI Magazine*. 20(1):27-41.

Kivlin, J. and F. Fliegel. 1967. Differential Perceptions of Innovations and Rate Adoption. *Rural Sociology*. 32(1):78-91.

Klein, M. and S. C.-Y. Lu. 1990. Conflict resolution in cooperative design. *The International Journal for AI in Engineering*, 4(4):168-180.

Kolodner, J.L. 1984. *Retrieval and Organizational Strategies in Conceptual Memory: A Computer Model*. Lawrence Erlbaum, Hillsdale, New Jersey.

Korsching, P.F., C.W. Stofferahn, P.J. Nowak and D. Wagener. 1983. Adoption characteristics and adoption patterns of minimum tillage: Implications for soil conservation programs. *Journal of Soil and Water Conservation*. 38:428-430.

Korsching, P.F. and T.J. Hoban. 1990. Relationships Between Information Sources and Farmers' Conservation Perceptions and Behavior. *Society and Natural Resources*. 3(4):1-10.

Kotz, A.M., K.R. Dittrich and J.A. Mülle. 1988. Supporting Semantic Rules by a Generalized Event/Trigger Mechanism in Advance in Database Technology. EDBT. Venice, Italy, pp. 76-91.

Kunnathur, A.S., M.U. Ahmed, and R.J.S. Charles. 1996. Expert Systems Adoption: An Analytical Study of Managerial Issues and Concerns. *Information and Management*, 3(1):15-25.

Lafond G.P. and R.J. Baker. 1986. Effects of Genotype and Seed Size on Speed of Emergence and Seedling Vigor in 9 Spring Wheat Cultivars, *Crop Science*. 26:341-346.

- Lafond, G.P., R.P. Zentner, R. Geremia and D.A. Derksen. 1993. The Effects of Tillage Systems on the Economic Performance of Spring Wheat, Winter Wheat, Flax, and Field Pea Production in East Central Saskatchewan. *Canadian Journal of Plant Science*. 73:47-54.
- Lamble, W. 1984. Diffusion and Adoption of Innovations. pp. 32-41. *In: D.J. Blackburn (Ed.) Extension Handbook*. University of Guelph, Guelph, Ontario.
- Lander, S.E. and V.R. Lesser. 1989a. A Framework for Cooperative Problem-Solving Among Knowledge-Based Systems. pp. 1-14. *Proceedings of the 1989 MIT-JSME Workshop on Cooperative Product Development*.
- Lander, S.E. and V.R. Lesser. 1989b. A Framework for the Integration of Cooperative Knowledge-Based Systems. pp. 472-477. *In: Proceedings of the 1989 IEEE International Symposium on Intelligent Control*. IEEE Computer Society Press, Washington, DC.
- Lander, S.E., V.R. Lesser and M.E. Connell. 1991. Conflict Resolution Strategies for Cooperating Expert Agents. pp. 183-200. *In: Proceedings of the 1990 International Working Conference on Cooperating Knowledge Based Systems*. Springer-Verlag, Berlin, Germany.
- Lander, S.E. and V.R. Lesser. 1992. Customizing Distributed Search Among Agents with Heterogeneous Knowledge. pp. 335-344. *In: Proceedings of the 1992 First International Conference on Information and Knowledge Management*.
- Lander, S.E. and V.R. Lesser. 1993. Understanding the Role of Negotiation in Distributed Search Among Heterogeneous Agents. pp. 438-444. *In: Proceedings of the 1993 International Joint Conference on Artificial Intelligence*.
- Lander, S.E. 1994. Distributed Search and Conflict Management Among Reusable Heterogeneous Agents. doctoral dissertation. University of Massachusetts, Amherst, Massachusetts.
- Lander, S.E. and V.R. Lesser. 1994. Sharing Meta-Information to Guide Cooperative Search among Heterogeneous Reusable Agents. Tech. Report CMPSCI 94-48, Department of Computer Science, University of Massachusetts, Amherst, Massachusetts.
- Lander, S.E. 1995. Agent-Based Integration of Legacy and Reusable Software Systems. Tech. Report SBIR/DMII. Blackboard Technology Group, Inc. Amherst, Massachusetts, July.
- Lander, S.E., S.M. Staley and D.D. Corkill. 1996. Designing Integrated Engineering Environments: Blackboard-Based Integration of Design and Analysis Tools. *In: Concurrent Engineering: Research and Applications, Special Issue on Multi-agent Systems in Concurrent Engineering*.
- Lee, L.K. 1983. Land tenure and adoption of conservation tillage. *Journal of Soil and Water Conservation*. pp. 166-168.
- Lee, L.K. and W.H. Stewart. 1983. Land ownership and the Adoption of Minimum Tillage. *American Journal of Agricultural Economics*. 65(2):256-264.

- Lee, L.K. and W.H. Stewart. 1985. Land ownership and the Adoption of Minimum Tillage: Reply. *American Journal of Agricultural Economics*. 65(3): 682-683.
- Lesser, V.R. and L.D. Erman. 1980. Distributed Interpretation: A Model and Experiment. *IEEE Transactions on Computers*. 29(12):1144-1163.
- Lesser, V.R. and D.D. Corkill. 1981. Functionally accurate cooperative distributed systems. *IEEE Transactions on Systems, Man and Cybernetics*.
- Lindner, R.K., A.J. Fischer and P. Pardey. 1979. The Time to Adoption. Mimeographed, University of Adelaide, Adelaide, Australia.
- Lindner, R.K. 1980. Farm Size and the Time Lag to Adoption of a Scale Neutral Innovation. Mimeographed, University of Adelaide, Adelaide, Australia.
- Lindner, R.K. and A.J. Fischer, 1981. Risk Aversion, Information Quality, and the Innovation Adoption Time Lag. Mimeographed, University of Adelaide, Adelaide, Australia.
- Lindwall, C.W. 1995. PARI: Parkland Agriculture Research Initiative. pp. 26-28. *In*: G.P. Lafond, H.M. Plas and E.G. Smith (Eds.) *Proceedings of the Workshop Bringing Conservation Technology to the Farm*.
- Lionberger, H.F. 1960. *Adoption of New Ideas and Practices*. Iowa University Press, Ames, Iowa.
- Lobb, D.A., R.G. Kachanoski and M.H. Miller. 1995. Tillage translocation and tillage erosion on shoulder slope landscape positions measured using ¹³⁷Cs as a tracer. *Canadian Journal of Soil Science*. 75(2).
- Lynne, G.D., J.S. Shonkwiler and R.R. Leandro. 1988. Attitudes and Farmer Conservation Behavior. *American Journal of Agricultural Economics*. pp. 12-19.
- Lyster, B. 1997. Check for Compatibility when you Shop for GPS Equipment. *In*: *Wheat, Oats & Barley Guide*, 116(1).
- Mallen, J. and M. Bramer. CUPID - An Interactive Knowledge Discovery Framework. University of Portsmouth, United Kingdom.
- Mann, C.K. 1977. Factors Affecting Farmers' Adoption of New Production Technology: Clusters of Practices. *In*: *Proceedings of the Fourth Regional Winter Cereals Workshop—Barley*, Amman, Jordan. pp. 24-28.
- Margolis, M. 1977. Historical Perspectives on frontier agriculture as an adaptive strategy. *American Ethnologist*. 4(1):42-64.
- Mason, R.L., L. Boersma and G.D. Faulkenberry. 1988. The use of open and closed questions to identify holders of crystallized attitudes: The Case of Adoption of Erosion-Control Practices Among Farmers. *Rural Sociology*. 53(1):96-109.
- McClelland, D.C. and D.G. Winter. 1969. *Motivating Economic Achievement*. Free Press, New York.

- McConkey, B.D., R.P. Zentner and W. Nicholaichuk. 1990. Perennial Grass Windbreaks for Continuous Wheat Production on the Canadian Prairies. *Journal of Soil and Water Conservation*, 45:482-485.
- McIntosh, W.A., J.K. Thomas and D.E. Albrecht. 1990. A Weberian Perspective on the Adoption of Value Rational Technology. *Social Science Quarterly*. 71(4):848-860.
- McKell, D. 1997. Precision Farming and Soil Conservation. *In: Prairie Steward - Farming For Your Future Environment*. (20).
- Medeiros, C.B. and P. Pfeffer. 1990. A mechanism for Managing Rules in an Object-oriented Database. Tech. Report Altair.
- Morgenstern, M. 1984. Constraint Equations: Declarative Expression of Constraints with Automatic Enforcement. pp. 153-299. *In: Proceedings of the 1984 International Conference on VLDB*.
- Napier, T.L. and D.L. Forster. 1982. Farmer Attitudes and Behavior Associated with Soil Erosion Control. pp. 137-150. *In: H.G. Halcrow, E.O. Heady and M.L. Cotner (Eds.) Soil Conservation Policies, Institutions and Incentives*. Soil Conservation Society of America, Ankeny, Iowa.
- Napier, T.L., C.S. Thraen, A. Gore and W. R. Gore. 1984. Factors Affecting Adoption of Conventional and Conservation Tillage Practices in Ohio. *Journal of Soil and Water Conservation*. 39(3):205-209.
- Napier, T.L., S.M. Camboni and C.S. Thraen. 1986. Environmental Concern and the Adoption of Farm Technologies. *Journal of Soil and Water Conservation*. 41(2):109-113.
- Neches, R., R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator and W.R. Swartout.. 1991. Enabling Technology for Knowledge Sharing. *AI Magazine*. 12(3):36-56.
- Nii, H.P. 1986. Blackboard Systems: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures. *AI Magazine*. 7(3):38-53.
- Nowak, P. 1983. Obstacles to adoption of conservation tillage. *Journal of Soil and Water Conservation*. pp. 162-165.
- Nowak, P. 1987. The Adoption of Agriculture Conservation Technologies: Economic and Diffusion Explanations. *Rural Sociology*. 52(2): 208-220.
- Nowak, P.J., P.F. Korsching, D.J. Wagener and T. Hoban. 1983. Sociological factors in the adoption of best management practices. Sociological Report, Iowa State University, Ames, Iowa.
- O'Donovan, J.T., et al. 1985. Influence of the Relative Time of Emergence of Wild Oat (*Avena Fatua*) on Yield Loss in Barley (*Hordeum Vulgare*) and Wheat (*Triticum Aestivum*). *Weed Science*. 33:48-53.
- O'Hare, G.M.P. and N.R. Jennings. (Eds.) 1996. *Foundations of Distributed Artificial Intelligence*. John Wiley & Sons, Inc., New York, NY.

- O'Neill, D. 1985. Farm Information Preferences. Saskatchewan Wheat Pool, Regina, Saskatchewan.
- Ozsu, M.T. and J.A. Blakeley. 1994. Query Processing in Object-Oriented Database Systems. pp. 146-174. *In*: W. Kim (Ed.) Modern Database Management - Object-Oriented and Multidatabase Technologies. Addison-Wesley/ACM Press.
- PAMI (Prairie Agricultural Machinery Institute). 1999. Direct Seeding Manual, A Farming System for the New Millennium. Humbolt, SK.
- Pampel, F. Jr. and John C. van Es. 1977. Environmental Quality and Issues of Adoption Research. *Rural Sociology*. 42:57-71.
- Parsave, K., M. Chignell, S. Khoshafian and H. Wong. 1989. Intelligent DataBases: Object-Oriented, Deductive Hypermedia Technologies. John Wiley & Sons Inc.
- Pavlik, C., E. Smith and C.W. Lindwall. 1995. PARMS - Planter and Residue Management System. (v2.1) [Computer software]. Agriculture and Agri-Food Canada, Lethbridge, Alberta.
- Perrin, R. and D. Winkelmann. 1976. Impediments to Tech. Progress on Small versus Large Farms. *American Journal of Agricultural Economics*. 58: 888-894.
- Perrin, R.K., D.L. Winkelmann, E.R. Moscardi and J.R. Anderson. 1976. From Agronomic Data to Farmer Recommendations: An Economics Training Manual. CIMMYT Information Bulletin No. 27. Mexico City, Mexico.
- Pimentel, D., E.C. Terhune, R. Dyson-Hudson, S. Rochereau, R. Samis, E.A. Smith, D. Denman, D. Reifschneider and M. Shepart. 1976. Land Degradation: Effects on food and energy resources. *Science*. 194:149-155.
- Polat, F., S. Shekhar and H.A. Guvenir. 1993a. Distributed conflict resolution among cooperating expert systems. *Expert Systems*. 10(4):227-236.
- Polat, F., S. Shekhar and H.A. Guvenir. 1993b. A Negotiation Platform for Cooperating Multi-agent Systems. *Concurrent Engineering: Research and Applications*. 1(3):179-187.
- Pool, I., de Sola and M. Kochen. 1978. Contacts and Influences. *Social Networks*. 1:5-51.
- Presser, H.A. and H.M. Russell. 1965. Acceptance of research results by farmers. *Review of Marketing and Agricultural Economics*. 33:147-165.
- Pringle, E.A. 1997. Tillage Erosion. [On-line] Available: www.res.agr.ca/cansis/publications/health/c7-3.html. Agriculture and Agri-Food Canada, Guelph, Ontario.
- Prundeanu, J. and P.J. Zerman. 1958. An evaluation of some economic factors and farmer's attitudes that may influence acceptance of soil conservation practices. *Journal of Farm Economics*. 40:903-914.

- Rahm, M.R. and W.E. Huffman. 1984. The Adoption of Reduced Tillage: The Role of Human Capital and Other Variables. *American Journal of Agricultural Economics*. 66:405-413.
- Rasmussen, W.D. 1982. History of soil conservation, institutions and incentives. pp. 3-18. *In*: H.G. Halcrow, E.O. Heady, and M.L. Cotner (Eds.) *Soil Conservation Policies, Institutions and Incentives*. Soil Conservation Society of America, Ankeny, Iowa.
- Rauniyar, G.P. and F.M. Goode. 1992. Technology adoption on small farms. *World-Development*. Pergamon Press, Inc., Tarrytown, N.Y. 20(2):275-282.
- Reitz, L.P. 1962. The Improvement of Wheat. *In*: A. Stefferud (Ed.) *After a Hundred Years*. United States Department of Agriculture, The United States Printing Office, Washington, D.C.
- Rieck, R.E. and G.C. Pulver. 1962. An Empirical Measure of Decision Making in Evaluating Farm and Home Development in Wisconsin. Agricultural Experiment Station and Cooperative Extension Service, Madison, Wisconsin. Research Bulletin 238.
- Roda, C., N.R. Jennings, and E.H. Mamdani. 1991. ARCHON: A Cooperative Framework for Industrial Process Control in Cooperating Knowledge Based Systems. Springer-Verlag.
- Rogers, E.M. and F.F. Shoemaker. 1968. *Communication of Innovations: A Cross-cultural Approach*. The Free Press, New York.
- Rogers, E.M. and L. Svenning. 1969. *Modernization Among Peasants*. Holt, Rinehart, and Winston, New York.
- Rogers, E.M. 1976. Where Are We in Understanding the Diffusion of Innovations? pp. 205-222. *In*: W. Schramm and D. Lerner (Eds.) *Communication and Change*. University of Hawaii Press, Honolulu, Hawaii.
- Rogers, E.M. 1983. *Diffusion of Innovations*. Macmillan Publishing Co., Inc., New York.
- Roling, N.G. 1985. Whither Rural Extension. Public lecture at the University of Guelph.
- Rosenschein, J.S., M. Ginsberg and M.R. Genesereth. 1986. Cooperation Without Communication. pp. 51-57. *In*: *Proceedings of the 1986 Conference for the American Association for Artificial Intelligence*.
- Rosenschein, J.S., 1987. Formal Theories of Knowledge in AI and Robotics. Tech. Report CSLI-87-84. Center for the Study of Language and Information, Stanford, California.
- Rumbaugh, J., M. Blaha, W. Premerlani, F. Eddy and W. Lorensen. 1991. *Object-Oriented Modeling and Design*. Prentice-Hall Publishers, Englewood Cliffs, New Jersey.
- Sampson, R.N. 1985. Government and conservation: Structuring an improved public role. pp. 1-6. *In*: *Soil Conservation: What should be the role of Government?* Indiana Cooperative Extension Service, Purdue University, West Lafayette.

- Schoney, R.A. in press. The Use of Costs of Production in Teaching, Research and Extension—A Case Example of 1987-95 Saskatchewan Wheat Production Costs.
- Schwartz, D.G. 1995. Cooperating Heterogeneous Systems. Kluwer Academic Publishers, Norwell, Massachusetts.
- Shaffer, M.J. and M.K. Brodahl. 1998. Rule-based management for simulation in agricultural decision support systems. *Computers and Electronics in Agriculture*. 21:135-152.
- Shelton, D.P., E.C. Dickey, P.J. Jasa, M.C. Hirschi and L.C. Brown. 1992. Water Erosion. *In: Conservation Tillage Systems and Management - Crop Residue Management with No-till, Ridge-till, Mulch-till*. Agricultural Engineering, North Dakota State University, Document MWPS-45.
- Sheng, O.R.L. and C. Wei. 1992. Object-Oriented Modeling and Design of Coupled Knowledge-base/Database Systems. pp. 98-105. *In: Proceedings of the 8th International Conference on Data Engineering*.
- Shepard, P. 1967. *Man in the Landscape*. Knopf, New York.
- Smith, E., C.W. Lindwall and C. Pavlik. 1995. *PARMS Users Manual*. Agriculture and Agri-Food Canada, Lethbridge, Alberta.
- Smith, E.G., T.L. Peters, R.E. Blackshaw, C.W. Lindwall, and F.J. Larney. 1996. Economics of Reduced Tillage Fallow-Crop Systems in the Dark Brown Soil Zone of Alberta. *Canadian Journal of Plant Science*. 76:411-416.
- Smith, P. 1987. Changes in Off-farm Work and Off-farm Income in Saskatchewan. pp. 167-172. *In: Proceedings of the Conference on Canadian Agricultural Outlook*.
- Smith, R.G. and R. Davis. 1981. Frameworks for Cooperation in Distributed Problem Solving. *IEEE Transactions on Systems, Man and Cybernetics*. 11(1):61-70.
- Soil Technologies Corp. Farm Product Summaries. [On-line] Available: www.206.26.90.8/soiltech/fprsum.html. Accessed: March 4, 1997.
- Staub, W.J. and M. Blase. 1974. Induced Technological Change in Developing Agriculture: Implications for Income Distribution and Agricultural Development. *Journal of Developing Areas*. 8:581-596.
- Stefferd, A. (Ed.) 1960. Power to Produce. *In: The Yearbook of Agriculture*. United States Department of Agriculture, The United States Printing Office, Washington, D.C.
- Stefferd, A. (Ed.) 1962. After a Hundred Years. United States Department of Agriculture, The United States Printing Office, Washington, D.C.
- Stonebraker, M., A. Jhingram, J. Goli and S. Potamianos. 1990. On rules, procedures, caching and views in database systems. pp. 281-290. *In: Proceedings of the ACM SIGMOD*.
- Stonebraker, M., L.A. Rowe, B. Lindsay, J. Gray, M. Carey, M. Brodie, P. Bernstein and D. Beech. 1991. Third-Generation Database System Manifesto. pp. 495-511. *In: R.A.*

- Meersman, W. Kent and S. Khosla (Eds.) Object-Oriented Databases: Analysis, Design, and Construction. Elsevier Science Publishers B.V. North-Holland.
- Su, S.Y.W. and A.M. Alashqur. 1991. A Pattern-Based Constraint Specification Language for Object-Oriented Databases. pp. 522-531. *In: Proceedings of the 36th IEEE Computer Society International Compcon Conference on Intellectual Leverage*. San Francisco, California.
- Swanson, L., S. Camboni and T. Napier. 1986. Barriers to Adoption of Soil Conservation Practices on Farms. pp. 108-120. *In: Lovejoy and T. Napier (Eds.) Conserving Soil: Insights from Socioeconomic Research*. Soil Conservation Society of America, Ankeny, Iowa.
- Sycara, K.P. 1985. Arguments of Persuasion in Labor Mediation. pp. 294-296. *In: Proceedings of the 1985 International Joint Conference on Artificial Intelligence*. Los Angeles, California.
- Sycara, K.P., et al. 1991. Distributed constrained heuristic search. *IEEE Transactions on Systems, Man and Cybernetics*. 21(6):1446-1461.
- Sycara, K.P. 1993. Machine learning for intelligent support of conflict resolution. *Decision Support Systems*, 10(2):121-136.
- Sycara, K.P. 1998. Multiagent Systems. *AI Magazine*. 19(2):79-92.
- Taylor, D. and W. Miller. 1978. The Adoption Process and Environmental Innovations: A Case Study of a Government Project. *Rural Sociology*. 43(4):634-648.
- Tekauz, A. 1997. Fusarium Head Blight of Cereals in Western Canada - A 10-year Retrospective. Cereal Research Centre, Agriculture and Agri-Food Canada, Winnipeg, Manitoba.
- Tenney, R.R. and N.R. Sandell Jr. 1981. Strategies for Distributed Decision making. *IEEE Transactions on Systems, Man and Cybernetics*. 11(8):527-538.
- Thomas, J.K., Ladewig and W.A. McIntosh. 1990. The Adoption of Integrated Pest Management Practices Among Texas Cotton Growers. *Rural Sociology*. 55:395-410.
- Thomsen, J. 1985. A Review of Agricultural Technology Transfer Policy Programs and Instruments. Working paper prepared for Regional Development Branch, Agriculture Canada, Ottawa.
- Tweeten, L. 1995. The structure of agriculture: Implications for soil and water. *Soil and Water Conservation*.
- UTI. 1997a. Market Assessment, Interim Report. Prepared for Agriculture and Agri-Food Canada. University Technologies International, Inc.
- UTI 1997b. PARI DSS'97 Market Assessment, Final Report. Prepared for Agriculture and Agri-Food Canada. University Technologies International, Inc.
- van Es, J.C. 1978. The Adoption/Diffusion Tradition Applied to Resource Conservation: Inappropriate use of Existing Knowledge. *The Rural Sociologist*. 3(2):76-82.

- Vines C.A. and M.A. Anderson. 1976. Heritage Horizons: Extension's Commitment to People. *In: Journal of Extension*, Madison, Wisconsin.
- Webster, B.F. 1995. Pitfalls of Object-Oriented Development. M&T Books, New York.
- Warriner, G.K. and T.M. Moul. 1992. Kinship and Personal Communication Network Influences on the Adoption of Agriculture Conservation Technology. *Journal of Rural Studies*. 8(3):279-291.
- Weisensel, W.P., A. Schmitz and W.H. Furtan. 1990. The Saskatchewan all risk crop insurance program: An evaluation of land use, technology adoption and participation. Saskatchewan Agriculture Development Fund, xii.
- Weiss, G. (Ed.) 1994. Multiagent Systems, A Modern Approach to Distributed Artificial Intelligence, MIT Press, Cambridge, Massachusetts.
- Werkman, K.J., S.J. Wagaman, D.J. Hillman, M. Barone and J.L. Wilson 1990. Design and Fabrication Problem-Solving Through Cooperative Agents. Report No. 90-05. ATLSS Engineering Research Centre, Lehigh University, Bethlehem, PA.
- Western Producer. 1995. Prairie ecosystem study looks at land use. *The Western Producer*, March 16.
- Wettlaufer, R.J. and P.B. Brand. 1992. Adoption of Soil Conservation Practices on the Canadian Prairies. pp. 397-411. *In: An International Perspective on the Socioeconomics of Soil and Water Conservation*. Honolulu, Hawaii.
- Whale, W.B., G. Hass, and B. Hobin. 1984. Agricultural Knowledge Exchange and Farmer Decision-Making. Unpublished manuscript, Division of Extension and Community Relations, University of Saskatchewan, Saskatoon, Saskatchewan.
- Whale, W.B. 1985. Communicating Agricultural Information - A Review of Recent Relevant Literature. Unpublished manuscript, University of Saskatchewan, Saskatoon, Saskatchewan.
- Wiederhold, G. (Ed.) 1996. Intelligent Integration of Information. Kluwer Academic Publishing, Boston, Massachusetts.
- Wilkening, E.A. and D. Johnson. 1961. Goals in Farm-decision Making as Related to Practice Adoption. Agricultural Experiment Station, Madison, Wisconsin, Research Bulletin 225.
- Winograd, T. and F. Flores. 1986. Understanding Computers and Cognition. Ablex. Norwood, New Jersey.
- Zentner, R.P. and C.A. Campbell. 1988. First 18 years of a Long-Term Crop Rotation Study in Southwestern Saskatchewan - Yields, Grain, Protein, and Economic Performance. *Canadian Journal of Plant Science*. 68:1-21.
- Zentner, R.P., S.A. Brandt, K.J. Kirkland, C.A. Campbell and G.J. Sonntag. 1992. Economics of Rotation and Tillage Systems for the Dark Brown Soil Zone of the Canadian Prairies. *Soil and Tillage. Res.* 24: 271-284.

Zentner, R.P., S.A. Brandt and C.A. Campbell. 1996a. Economics of Monoculture Cereal and Mixed Oilseed-Cereal Rotations in West-Central Saskatchewan. *Canadian Journal of Plant Science*, 76:393-400.

Zentner, R.P., B.G. McConkey, C.A. Campbell, F.H. Dyck and F. Selles.. 1996b. Economics of Conservation tillage in the Semi-Arid Prairie. *Canadian Journal of Plant Science*. 76:697-705.

Appendices

APPENDIX A

Summary of Important Producer Issues for Conservation Cropping Decisions as Perceived by the PARI DSS Steering Committee 18 February 1993

Note: Individual committee member's opinions have been grouped into categories to help summarize the diverse number of opinions.

<u>Rank</u>	<u>Issue</u>	<u>Important Aspects</u>
1	Weed management	<ul style="list-style-type: none">• weed ID: control: mechanical, herbicide, biological with crop rotation?• herbicide selection: weeds present and density, crop (application) costs, recropping restrictions, timing (pre/post harvest, pre-seeding), pre emerge vs. post emerge herbicides• more chemicals required? greater cost? increased chemical residue?
2	Crop rotation/selection/mangmt	<ul style="list-style-type: none">• rotation with regards to weeds, insects, plant diseases• selection as it relates to agro-climatic zones• plant nutrition related to variety performance, plant health, and sustainability of soil, increase organic matter• economic reasons (e.g., including pulse crop and canola to improve \$• Plant arrangement, row spacing• crop selection: with production requirements, which has market and profit
3	Production system/yield/erosion	<ul style="list-style-type: none">• effect of cropping practice on potential water and erosion• production potential as function of soil type and agro-ecological zone• producers level of commitment to conservation farming (flexibility)• diversification: economics and agronomics of alternative cropping systems• nature and rate of change from "steady state" under conventional till to "steady state" under no tillage• conservation, crop productivity, soil degradation

<u>Rank</u>	<u>Issue</u>	<u>Important Aspects</u>
4	Economics	<ul style="list-style-type: none"> • overall system evaluation with links to concerned areas • evaluate economic indicators (debt/asset, cash/fixed costs, benefits/costs of specific actions), net income, stability • short/long term returns to investment & risk associated with new production practices • best suit of land base (grain, forage, livestock) • modify existing operation for conservation practices at minimal capital cost
5	Fertility/nutrients	<ul style="list-style-type: none"> • fertilizer placement (NH₃ at time of seeding) • seed placed vs. banded phosphate, potassium and sulfur • side banded or banded with seed with adequate seed/fertilizer separation • alternate sources of nitrogen, expectations when used, residual effects • reduction in yield caused by erosion? nutrients to compensate this loss?
6	Tillage	<ul style="list-style-type: none"> • Standard tillage definitions, learn more re whole cropping sequences • techniques for minimum, reduced, and zero tillage • each tillage operation: useful?, what accomplished?, other solutions? • difference in yield and net profit on different soil types with different tillage practices
7	Crop residue management	<ul style="list-style-type: none"> • cropping practice: rotation, mowing, tillage (deep, rotary harrow) • fall management of residue: straw/chaff spreading and/or collection • appropriate seeding equipment
8	Information	<ul style="list-style-type: none"> • production: appropriate, relevant, timely, sources-producers/advisors • markets (where to sell, contracts), markets for new crops • easy access to credible, knowledgeable expertise • field level information including historical information on soils and crop performance
9	Equipment/machinery selection	<ul style="list-style-type: none"> • tillage system, farm size, climate, soil texture • modifications that can be done to existing machinery

<u>Rank</u>	<u>Issue</u>	<u>Important Aspects</u>
10	Insect management	<ul style="list-style-type: none"> • insect ID & control
11	Plant diseases	<ul style="list-style-type: none"> • especially residue related
12	Moisture	<ul style="list-style-type: none"> • utilization (various crops on different soil types), collection, retention, interaction with fertility requirements
13	Government support/policy	<ul style="list-style-type: none"> • impact on new production practices and choice of crops
14	Climate/weather data	<ul style="list-style-type: none"> • probabilities of events (e.g. precipitation), conditional probabilities of events based on previous history
15	Soils data/analysis/landscape	<ul style="list-style-type: none"> • level of nutrients required to achieve target yield • how to treat changes in slope and fertility on a landscape basis
16	Knowledge, education	<ul style="list-style-type: none"> • basic: biology, plant science, animal science, soil science, farm business management, computer literacy
17	Harvesting	<ul style="list-style-type: none"> • straight cut vs. swathing

APPENDIX B

Software Evaluation Report of Crop Planning Module in Farm Smart 2000 V.0.7.2

by:

**Henry C. de Gooijer, B.S.A.
Producer at Kelliher, SK.**

Introduction:

An evaluation of Farm Smart 2000 (FS2000) was conducted in order to determine its abilities and capabilities in supplying pertinent information with regards to cropping decisions. This evaluation was not a comprehensive evaluation of all functions of the software, however it did strive to test for a core competency in providing useful crop production decision support with various levels of input data.

Evaluation Goals:

The main goals of the evaluation were to determine the operational capability of the Crop Planning Module in the following areas:

- 1) To accept farm and field scale data.
- 2) To evaluate the data provided.
- 3) To present agronomic and economically significant results.

The evaluation of the Basic Decision Support and Advanced Decision Support modules was not requested in this evaluation. However this does not diminish the importance of access to electronic information contained on the Farm Smart 2000 web site.

Assumptions:

All information entered into the record-keeping component was not actual field data but rather derived from experience. As such, selection of certain characteristics of the data such as crop rotation and physical attributes of the defined farm were made to test specific capabilities of the software in differentiating between differing conditions. The following assumptions were made in creating the dataset for the evaluation.

- 1) Climatic conditions are the same for all fields.

- 2) While all fields are indicated as a specific location, the agronomic information and physical characteristics indicated may not be accurate for the actual location specified.
- 3) All locations are in Saskatchewan.
- 4) Field records for the two fields are not complete, however sufficient data was provided to allow for the operation of the Crop Planning Module. Having less than complete historical records was determined to be a likely operating environment for many producers.

Record Keeping Evaluations:

Good decision making is based on having complete and accurate information upon which to base decisions. The role of any record keeping system, computerized or otherwise, is to provide an accurate record of previous activities upon which the success of those activities may be evaluated and future actions planned.

The Record Keeping Module (RKS) of FS2000 is an elaborate series of data tables and input screens with the capability to store detailed field information for a wide range of economic and agronomic practices and factors. While the capability exists to store detailed information, the structure is not excessively restrictive allowing a minimum amount of information to be stored and utilized within the record databases.

In order to test the RKS, a sample farm (Test Farm) with two fields (North Forty and Sandy Acres) was created with three years of historical data (1997-1999). A three year time frame was chosen for historical data in order to emulate mid length rotations and to test how previous cropping decisions may affect current and future decisions.

The field, North 40, was selected as a medium textured, relatively level landscaped parcel of land with few production restrictions. The other field, Sandy Acres (Figure B.1) was defined as a relatively light textured soil, with more restrictive conditions for annual crop growth and greater erosion potentials if not properly managed.

A rotation of cereal, pulse, cereal, and oilseed was entered into the historical records for each field. While some years of historical data were entered in some detail, most historical data was left quite sparse in order to test how the Crop Planning Module would react under minimum data conditions.

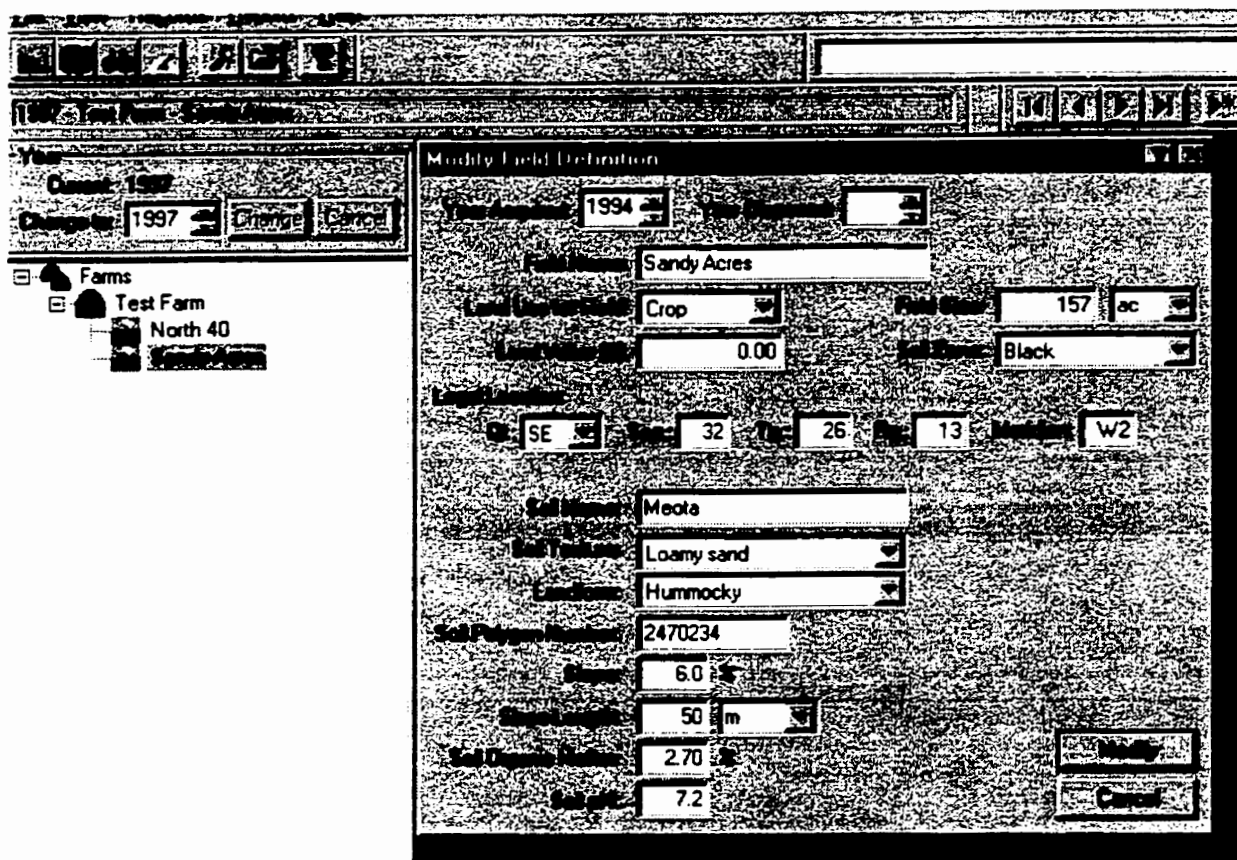


Figure B.1 Farm Field Definition Screen

A complement of Farm Equipment was created and defined for the farm (Figure B.2).

Some of the information required for the description of the equipment is quite detailed. But since farm equipment is entered at the farm level and not the field level, supplying the level of detail is not as onerous as if it had to be added for every field in every year.

When defining the farm operations as events it is sometimes difficult to determine between what are considered events which affect the entire farm and which affect a single field. This is especially true for Event Folders dealing with debt, taxes and rent. As this information is utilized nominally within the RKS and Crop Planning Module, minimum data was entered in these specific event folders. Greater emphasis was placed on populating the agronomic event folders (Figure B.3).

File View Report Search Database Window Help

Flexcol Airseeder

Year: 1997

Farms: Test Farm, North 40

Equipment 1

Equipment Name: Flexcol Airseeder

Machine Type: Airseeder - line type

Capacity: 75,000

Row Spacing: 1.0 in

Row Width: 30 in

Row Depth: 33.0 ft

Row Rate: 7.00 Imp Gal

Row Count: 750

Row Length: 10

Row Area: 112

Figure B.2 Equipment Editing Screen

File View Report Search Database Window Help

02/17/1997 - Wheat

Year: 1997

Farms: Test Farm, North 40

Event Name: Wheat

Event Type: 0.0 cm

Event Date: 02/17/1997

Event Time: 0

Event Location: On row

Event Area: 157 ac

Event Rate: 0.0 cm

Event Speed: 0.0 km/h

Event Yield: 0.00 bu

Event Loss: 0 bu

Figure B.3 Events Entry and Editing Screen

While detailed crop records are essential for good decision making and as a history of previous activities, there was very little additional benefit in creating highly detailed historical records in the perspective of being able to evaluate them in the Crop Planning module for future planning needs. Beyond the most recent year before the year being considered in the Crop Planning module, there is very little benefit in having more than the crop and yield indicated. The Crop Planning module is mainly utilized for forward planning, therefore most historical data becomes less important for future planning however it still provides a critical historical record of activities and results.

The Report Wizard within the RKS module provides an excellent capability to select and present information contained in the RKS database (Figure B.4).

Weed Name	Emergence Date	Stage	Density	
No data to report.				
1998				
Fertilizing Events				
Date	Fertilizer	Application Method	Rate	
No data to report.				
Harvest Events				
Crop Name	Grade of Crop	Yield	Harvest Date	Total Production
Peas		30.000 bu/ac	02/17/1998	4710.000 bu
Herbicide Application Events				
Herbicide Applied	Application Date	Herbicide Cost		
No data to report.				
Seeding Events				
Seeding Date	Crop Name	Variety	Area Seeded	Duration of Activity
02/17/1998	Peas	CARNEVAL	157 ac	0 hours
Weed Events				
Weed Name	Emergence Date	Stage	Density	
No data to report.				

Figure B.4 Example of Event Report from RKS module

Reports may also be presented in Summary or detail formats and specialized formats saved for future reference making the reporting function easy and useful. However report structures and titling require more refinement to make them more descriptive and useful.

Crop Planning Evaluations:

While good field records can provide a useful historical record for farm managers, the ability to utilize records to plan future crops and activities within the Crop Planning Module is very beneficial.

The Crop Planning Module employs a table like structure with tabs which allows the movement between input data and results. This design makes it quite easy to make quick changes to input data and perform evaluations of the data (Figure B.5).

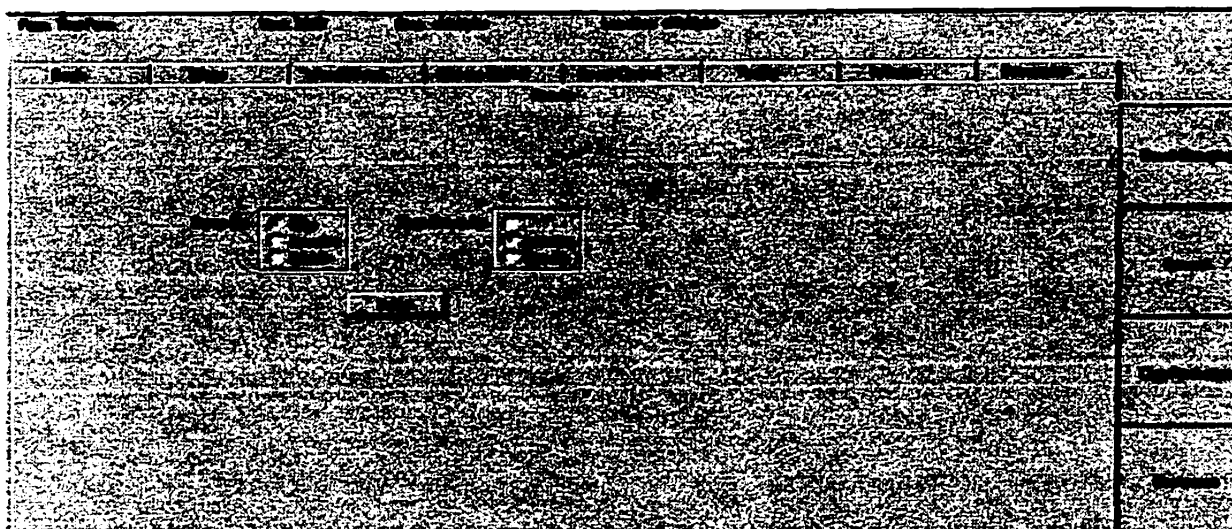


Figure B.5 Input/Results Structure of the Crop Planning Module

The matrix of cells created by the crop and field columns and rows provides an environment for easy selection of specific crop – field evaluations (Figure B.6). Selecting cells individually or by column or row simplifies data input and makes the evaluation of various crop planning scenarios relatively easy. Direct access to available software such as the Crop Protection Planner when entering pesticide information takes the guess work out of supplying reasonable estimates of cost and control while eliminating the need for additional reference materials.

While the Crop Planning module utilizes historical data, it also requires the entry of agronomic practices and events for the planning year in question. This information is easily entered under the appropriate tabs indicated in Figure B.7.

Information may be entered for tillage goals, tillage practices, weed control, disease control, fertility, weather conditions and economics. Once all information is entered for crops and fields, the required matrix of crop and field combinations are easily selected. Evaluations occur relatively quickly with the results displayed from the same tab structure utilized for data input (Figure B.8).

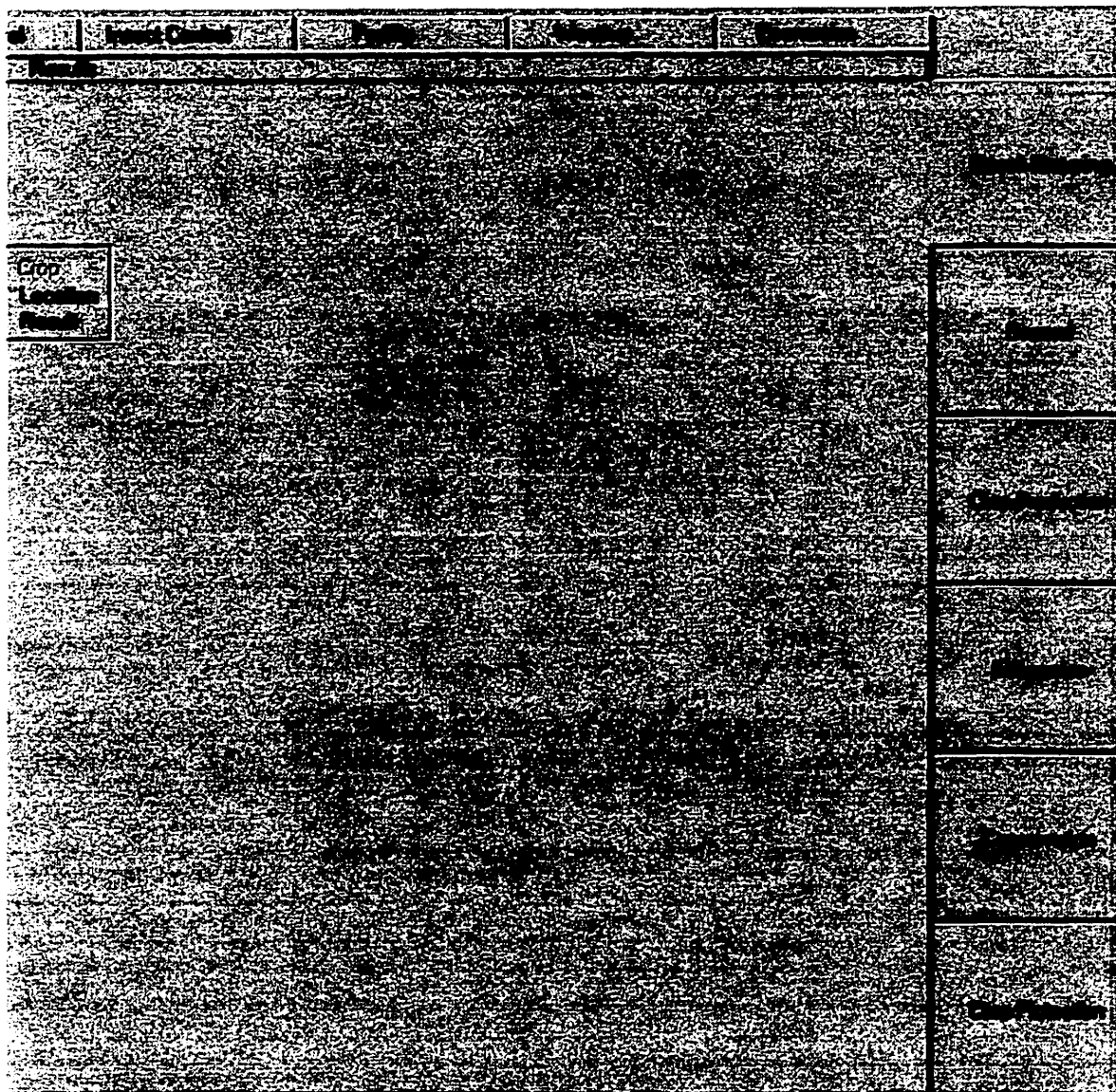


Figure B.8 Crop Planning Module Results Structure

Results may be organized in various manners but are presented under the categories of General results, Crop Production, Economics, Conservation and Crop Protection. The General Results tab provides a general summary of all results. (Figure B.9).

The general results folder allows a quick comparison of results. In the evaluation performed, either Flax or canola are to be grown on either one or both of the fields. The General Results folder shows how the options compare to one another.

The Crop Production Folder (Figure B.10) provides more detailed information on crop production factor including fertilization, yield results and expectations, seeding rates and depths and rotation concerns.

File Edit View Options Help

Form Test Farm Year 2000 Crop 000000 Location 000000

Sub Main Westland Northland Southland Planting Winter Economic

General Results

Crop Year: 2000
Report Generated: 02/21/2000
Farm: Test Farm

Canola (Q2)

North 40

Conservation Goal: Minimum Tillage
Erosion Risk: Low
Herbicide Residue Risk: None
Potential Yield: 36 bu/ac
Gross Return: \$ 222.06 /ac
Total Expenses: \$ 91.58 /ac

Net Return: \$ 130.48 /ac

Sandy Acres

Conservation Goal: Minimum Tillage
Erosion Risk: Low
Herbicide Residue Risk: None
Potential Yield: 26 bu/ac
Gross Return: \$ 160.58 /ac
Total Expenses: \$ 91.58 /ac

Net Return: \$ 69.00 /ac

Flax (SOMME)

North 40

Conservation Goal: Minimum Tillage
Erosion Risk: Low
Herbicide Residue Risk: None
Potential Yield: 26 bu/ac
Gross Return: \$ 150.76 /ac

Result Group
General
Crop Production
Economic
Conservation
Crop Production

Figure B.9 Crop Planning Module General Results Folder

File Edit View Options Help

Crop Production Results

Canola (Q2)

North 40

2001: To be determined
2000: Canola
1999: Wheat
1998: Peas
1997: Barley
1996:
1995:

Herbicide Residue Risks

Bioassay
Year Risk Required Herbicide

There are no herbicide residue risks.

Seeding Date: 05/09/2000 Harvest Date: 08/14/2000
Seeding Depth: 2.0 cm (Recommended: 2.0)
Seeding Rate: 7.00 lb/ac (Recommended: 5.00 - 7.00)
Seed Bed Temperature: Normal
Yield Target: 35 bu/ac Potential: 36 Variety Average: <Unknown> Break-Even: 15

Fertilizer Summary

(All values are in lb/ac)

Levels	Nitrogen	Phosphorus	Potassium	Sulfur
--------	----------	------------	-----------	--------

Figure B.10 Crop Planning Module Crop Production Folder.

The Economics Folder provides limited accounting of the costs associated with the planned agronomic activities (Figure B.11). Very little economic evaluation occurs. Cost comparisons are made between typical, and predicted costs (Figure B.12) to show how the current cropping plan compares to area average costs. More economic evaluation and information are required.

Economic Results			
Canola (Q2)			
North 40			
	Calculated	Typical	Maximum
All values are in \$/ac)			
Gross Return:	222.06	125.00	n/a
Costs:			
Seed:	14.07	6.00	20.00
Tillage:	8.00	9.70	15.00
Herbicide:	31.01	15.53	20.00
Fungicide:	0.00	2.25	10.00
Insecticide:	0.00	2.25	10.00
Fertilizer:	38.50	28.20	25.00
Total Costs:	91.58	63.93	100.00
Net Return:	130.48	61.07	n/a
Sandy Acres			

Figure B.11 Crop Planning Module Economics Folder

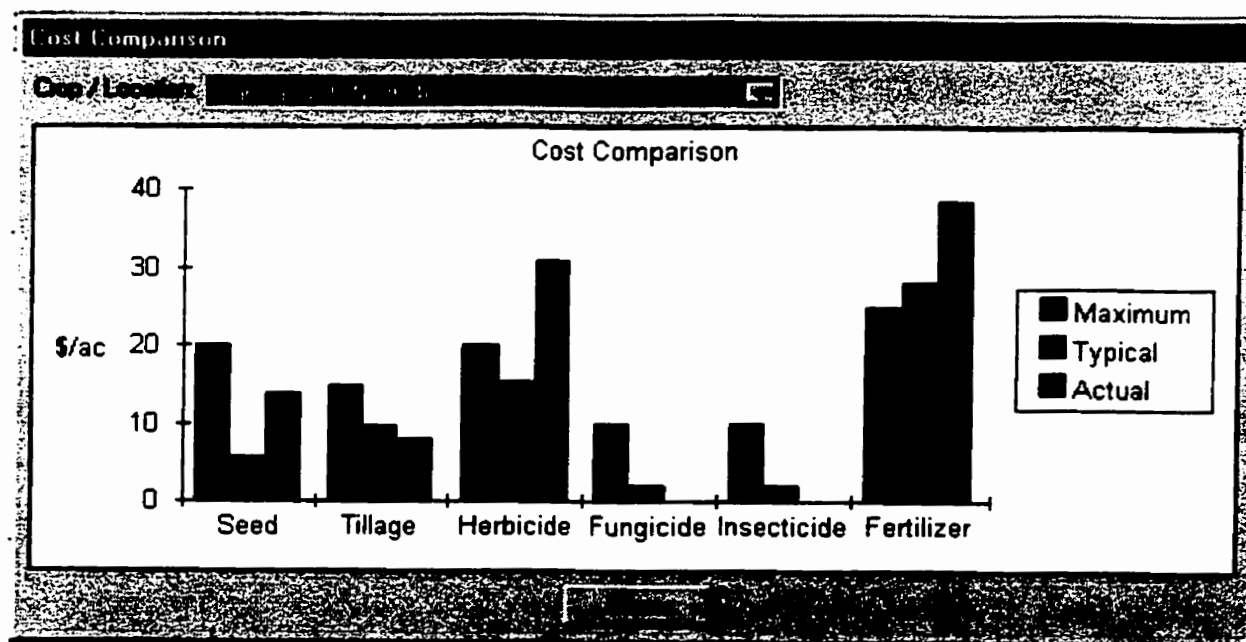


Figure B.12 Cost comparison graphs in Crop Planning Module

The Conservation module introduces a stewardship consideration to the evaluation which is otherwise economic or agronomic (Figure B.13). The Conservation module provides an organic matter and erosion risk evaluation for the field, if the planned cropping option is carried out.

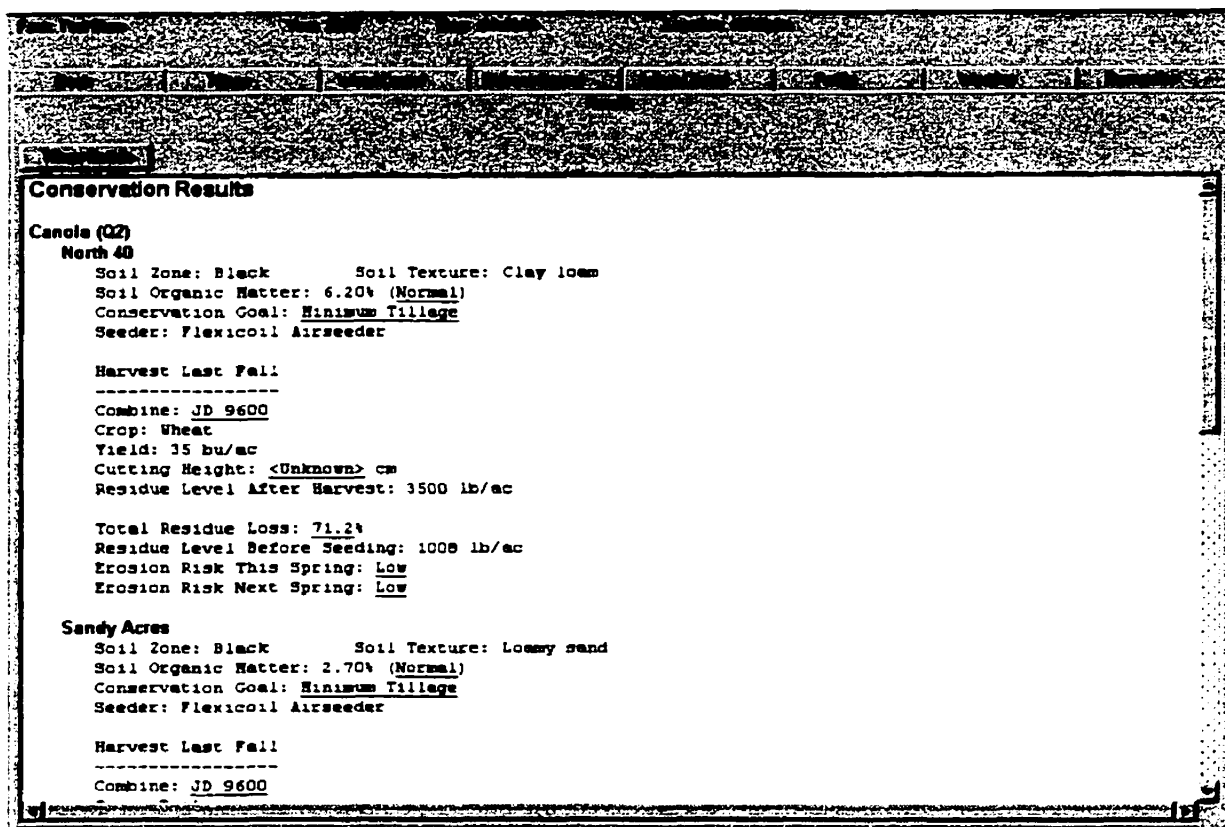


Figure B.13 Crop Planning Module, Conservation Results Folder.

The Crop Protection Module is dedicated to crop protection issues including weed information for hard to control weeds, herbicide costs, herbicide residues and weed resistance (Figure B.14).

Results and Expectations:

The Crop Planning module does not make specific recommendations or “decisions” for the user, nor is it required to, being a “decision support system”. It does however, make the user aware of the large number of factors considered in making cropping decisions, and presents those factors to the user so that the consequences of any decisions may be viewed in light of agronomic and economic considerations.

Herbicide Application Details

Herbicide Application Details

Canola (Q2)
North 40
Weed Resistance Problems: None

Weed(s): Wild Oats; Green Foxtail; Canada Thistle
 Herbicide: VENTURE 25DG + LONTREL 360
 Group(s): 1, 4
 Cost: \$31.01/ac entered; \$31.01 calculated
 Rates

Entered	Recommended
0.120	0.120 - 0.280 kg/ac
0.168	0.168 - 0.336 L/ac
0.000	0.000 /ac

This herbicide places restrictions on future cropping options.

Sandy Acres
Weed Resistance Problems: None

Weed(s): Wild Oats; Green Foxtail; Canada Thistle

Figure B.14 Crop Planning Module, Herbicide Details Window

The success and capability of the Crop Planning module may be expressed in terms of the evaluation goals listed above. It is able to store and utilize field record information, to evaluate the data, and to produce accurate agronomic and economic results based on my farming experience.

Farm Smart 2000 in its current state, cannot be considered a complete product but then with the rapid advancement of agricultural and computer technology one might never consider it complete. FS2000 provides an excellent framework upon which further development may occur. For instance, the Record Keeping System is capable of providing data storage requirements for a good field record keeping system and providing a solid base for additional evaluations and in turn more decision support.

The design of the Crop Planning Module is very workable and able in providing results to the user. It does provide useful information and evaluation in some agronomic areas however it is lacking in areas such as fertility and economic evaluations, and is thus limited in its capabilities to provide well rounded support information.

It may be unreasonable to expect that a decision support system working with a relatively limited dataset may provide exacting results to the users. Given such an environment, one can hardly expect exact quantitative results for comparisons. Results should be viewed in a more relative sense. A more realistic expectation for a decision support system is to provide management ideas and to flag or prevent management oversights which may lead to cropping disasters or poor performances. In that light, Farm Smart 2000 is effective in meeting such needs.